Mobile IP Movement Detection Optimisations in 802.11 Wireless LANs

Prepared by:
Albert M. Hasson

Supervised by:
Neco Ventura

Department of Electrical Engineering
University of Cape Town
2005

This dissertation is submitted to the University of Cape Town
in fulfilment of the academic requirements
for the Degree of Master of Science in Engineering

20 February 2005

Declaration

I declare that this thesis is my own work. Where collaboration with other people has taken place, or material generated by other researchers is included, the parties and/or material are indicated in the acknowledgements or references as appropriate.

This work is being submitted for the Master of Science Degree in Electrical Engineering at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

Albert M. Hasson Date

Acknowledgements

בס"ד

I would like to express my sincere gratitude to the following individuals and organisations for their assistance during the course of this project.

Mr. Neco Ventura, for his supervision and guidance throughout the project.

Sven Shepstone, to whom I am grateful for his advice, constructive feedback and constant encouragement.

The National Research Foundation (NRF) and the South African Department of Labour (DoL) for their generous financial contributions towards this research.

My fellow colleges in the Communications Research Group (CRG) and Speech Technology (STAR) group at UCT, for the useful and interesting discussions on mobility, life, the universe and everything.

Reviva, for being an amazing friend. Thanks for keeping up the wonderful distractions all this time.

My parents and sister, for their love and constant support in every aspect of my life. Thank you.

Synopsis

The IEEE 802.11 standard was developed to support the establishment of highly flexible wireless local area networks (wireless LANs). However, when an 802.11 mobile node moves from a wireless LAN on one IP network to a wireless LAN on a different network, an IP layer handoff occurs. During the handoff, the mobile node's IP settings must be updated in order to re-establish its IP connectivity at the new point of attachment. The Mobile IP protocol allows a mobile node to perform an IP handoff without breaking its active upper-layer sessions. Unfortunately, these handoffs introduce large latencies into a mobile node's traffic, during which packets are lost. As a result, the mobile node's upper-layer sessions and applications suffer significant disruptions due to this handoff latency. One of the main components of a Mobile IP handoff is the movement detection process, whereby a mobile node senses that it is attached to a new IP network. This procedure contributes significantly to the total Mobile IP handover latency and resulting disruption.

This study investigates different mechanisms that aim to lower movement detection delays and thereby improve Mobile IP performance. These mechanisms are considered specifically within the context of 802.11 wireless LANs. In general, a mobile node detects attachment to a new network when a periodic IP level broadcast (advertisement) is received from that network. It will be shown that the elimination of this dependence on periodic advertisements, and the reliance instead on external information from the 802.11 link layer, results in both faster and more efficient movement detection. Furthermore, a hybrid system is proposed that incorporates several techniques to ensure that movement detection performs reliably within a variety of different network configurations.

An evaluation framework is designed and implemented that supports the assessment of a wide range of movement detection mechanisms. This testbed allows Mobile IP handoffs to be analysed in detail, with specific focus on the movement detection process. The performance of several movement detection optimisations is compared using handoff latency and packet loss as metrics. The evaluation framework also supports real-time Voice over IP (VoIP) traffic. This is used to ascertain the effects that different movement detection techniques have on the output voice quality. These evaluations not only provide a quantitative performance analysis of these movement detection mechanisms, but also a qualitative assessment based on a VoIP application.

Contents

List of Figures

List of Tables

Glossary

This section defines some of the commonly used terms and abbreviations that appear throughout this document.

802.11 A wireless local area networking (wireless LAN) family of standards developed by the IEEE that emulates conventional Ethernet links. The 802.11 standards include several different physical implementations, namely 802.11a/b/g, which provide a range of data rates. 802.11 is also commonly known as Wi-Fi.

Access Point (AP) An entity that bridges information between the wireless medium and the distribution medium on behalf of its associated stations. Access points are only used in infrastructure wireless LANs.

Access Router (AR) An IPv6 router on the periphery of a network that provides mobility services to visiting mobile nodes.

Association The 802.11 association service establishes a mapping between a station and an access point. This allows a station to transmit and receive traffic over the distribution medium.

Authentication The 802.11 authentication service allows a wireless device to identify itself to other stations (such as an access point).

Basic Service Set (BSS) A set of 802.11 stations that are coordinated as a single unit.

Care-of address (CoA) A temporary IP address allocated to a mobile node while it is visiting a foreign network

Corresponding Node (CN) Any network node that is communicating with the mobile node. A corresponding node may be a conventional fixed host.

Extended Service Set (ESS) A set of one or more interconnected basic service sets (BSSs).

Extended Service Set Identifier (ESSID) An alphanumeric identifier assigned to a particular Extended Service Set. All stations within an ESS must share a common ESSID.

Foreign Agent (FA) An IPv4 router that supports mobility functions for visiting mobile nodes. For example, a foreign agent broadcasts periodic advertisements that allow visiting mobile nodes to detect their arrival on the network.

Home Agent (HA) An IPv4 or IPv6 router residing on a mobile node's home network. The home agent forwards a mobile node's traffic to its current point of attachment while it is away from its home network.

IP Subnet An IP subnetwork is a component network of a larger IP internetwork.

Link Entities that share a link are physically connected through a communication channel. These entities are able to communicate directly using a link layer protocol.

Medium Access Control (MAC) A protocol that governs how network nodes access a physical medium (CSMA/CD and Token Ring are examples of MAC protocols).

Mobile IP The Mobile IP protocol is a networking layer technology that allows a mobile node to migrate through different IP networks while maintaining its upper-layer sessions. Two versions of Mobile IP have been developed: Mobile IPv4 (MIPv4) and Mobile IPv6 (MIPv6).

Mobile Node (MN) A network node that is able to communicate while moving through different networks. Note that a "portable" device implies a device that may be transported easily.

Movement Detection IP layer mechanisms that allow a mobile node to detect its arrival on a new IP network, i.e. as part of a Mobile IP handoff.

Signal-to-noise ratio (SNR) The difference in decibels between the received signal strength and the noise level.

Station An IEEE 802.11 device that provides a network interface to the wireless medium. An 802.11 station incorporates both 802.11 medium access control (MAC) and physical layer entities.

Voice over IP (VoIP) The two-way transmission of voice information over a packet-switched TCP/IP network. (Also known as "IP telephony".)

Chapter 1

Introduction

1.1 Background Information

Traditionally, hosts connected to the Internet have been large and immobile desktop systems. In addition to their physical size, these hosts have been limited to accessing the Internet through fixed wired network interfaces. In recent times however, this scenario has rapidly begun to change. Firstly, technological advancements have allowed computing equipment to be miniaturised, paving the way for powerful portable devices. This has resulted in a wave of new portable computers, such as laptops and personal digital assistants (PDAs) emerging in the market place. However these portable devices initially had to rely on wired interfaces to communicate with other devices. The advent of wireless networking standards has given these portable devices the ability to become truly mobile. Wireless technologies based on these standards allow such devices to maintain network connectivity with other network nodes through small radio transceivers.

In 1997, the Institute of Electrical and Electronic Engineers (IEEE) released the 802.11 wireless networking standard. The 802.11 standard was designed to facilitate the creation of flexible wireless local area networks (wireless LANs). A wireless LAN is conceptually analogous to its wired counterpart, the LAN, and provides similar services and functionality. Originally, the first release of the 802.11 standard only supported 1 and 2 Mbps data rates, which proved to be insufficient for most conventional LAN requirements. However, in 1999 the IEEE ratified the 802.11b standard (also known as Wi-Fi) that supported higher data rates up to 11 Mbps. The IEEE has released several other standards as part of the 802.11 family, including 802.11a and 802.11g that use different modulation technologies

to support various data rates. The 802.11 family of standards have become increasingly popular because they have proved to be effective at extending or even replacing conventional wired LANs. The current wide-spread consumer interest being expressed in 802.11 devices is being fuelled by many additional factors. For example, the Wi-Fi Alliance[1] is a non-profit organisation formed in 1999, consisting of wireless systems manufacturers and companies, whose role it is to ensure that all 802.11 equipment is compatible and interoperable. The result has been that 802.11 technology is relatively easy and inexpensive to implement on a consumer level.

The higher data rates provided by the newer 802.11 standards have enabled the support of a number of new applications and services. In fact, 802.11 technology has given rise to the establishment of public wireless access networks, commonly known as hot spots. Hot spots provide 802.11-enabled terminals with high-speed wireless Internet access. Commercial hot spots are often located in busy urban areas such as airport lounges, coffee shops, and hotels while some universities have even established free public hot spots on their campuses. The number of these hot spots has been increasing dramatically in accordance with the increasing popularity of 802.11 wireless devices. In fact, the city of London contains over 6000 overlapping hot spots while in Seoul, South Korea about half of all Internet usage runs over these types of wireless networks [39].

The 802.11 hot spot is an example of how wireless LANs may be used to create versatile access networks. An 802.11 wireless LAN extends over a limited physical coverage area due to the characteristics and constraints of the underlying radio technology. 802.11 is therefore often referred to as a "last-meter" technology. Despite this, a wireless network node experiences a high degree of flexibility for movements within this coverage area (ignoring effects such as signal attenuation and blockage due to obstacles). The network node uses its 802.11 interface to establish a wireless link between itself and the rest of the access network infrastructure. The access network in turn provides the coupling between the wireless LAN and the wired Internet backbone. 802.11 is a link layer technology and is therefore responsible for establishing reliable wireless connections to transport upper-layer data. It does not deal with how this data should be transported through the Internet.

The Internet Protocol (IP) is the standard mechanism used to transfer information to remote hosts through the Internet. IP was originally developed as part of the TCP/IP protocol suite used in the ARPANET. The ARPANET was a research network, first established in the late 1960s, that linked universities and government agencies throughout

[1]Further information can be found at `http://www.wi-fi.org`

the United States. IP was the "glue" that allowed the diverse networks that existed at the time to be joined to form a much larger unified internetwork. IP also ensured that the ARPANET was robust, and that if individual nodes malfunctioned, sections of the network would not become isolated.

The ARPANET eventually evolved into today's world-wide Internet. Today, IP facilitates the interconnection of the heterogeneous networks that make up the Internet, as it did for the Internet's predecessors. IP serves as a network layer inter-networking protocol, designed to function independently from underlying hardware, physical media or data link technology. It allows data traffic to be routed through these subnetworks while trying to minimise the data loss that may result from congestion. IP is a packet-switched protocol where information is transmitted in discrete packets that are treated independently of one another. Each packet contains a destination and a source address field that allows it to be routed through the Internet and delivered to the destination host. Lastly, the IP protocol is a connectionless "best-effort" service that does not guarantee that all packets will be delivered to the destination or that they will be delivered in the correct order.

IP uses a hierarchical addressing scheme that makes the process of routing packets through a large internetwork scalable and efficient. An IP address is divided into a network part (network identifier) and a host part (host identifier). The network portion defines a particular IP subnetwork while the host section defines a specific host on this network. IP packets are routed through the Internet based solely on their destination address (and perhaps also the congestion state of the network). Usually, this is achieved by first routing a packet to the destination subnetwork, and then delivering the packet to the specific end host.

The characteristics of IP listed above illustrate that the TCP/IP protocol suite was developed assuming that the end points of a network would be fixed nodes, attached to the network at static points. Consequently, the current IP protocol does not include support for network-level mobility. However, the increasing numbers of portable devices equipped with wireless network interfaces is slowly challenging these underlying assumptions. The IP protocol's lack of support with regard to mobility stems from the fact that two aspects are associated with every IP address. Firstly, an IP address defines a network node's identity. Secondly, due to its hierarchical structure mentioned above, an IP address also implies the node's topological position.

These factors make network layer mobility using the standard IP protocol very difficult.

Figure 1.1 below will be used to illustrate these issues in greater detail. Two separate IP subnets are shown, both connected to the Internet. An IP network node (such as a laptop) is connected to subnet A and is communicating with a corresponding node (not shown). In order to transfer IP traffic through the Internet, the node must have a defined IP address. Other additional IP configuration settings are also necessary, such as the default gateway's IP address, but these are secondary.

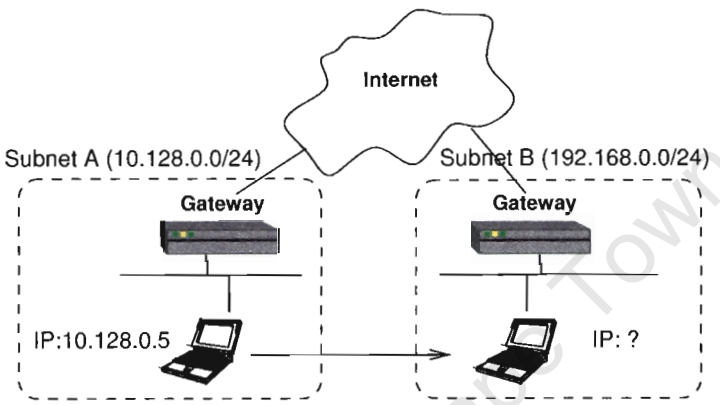

Figure 1.1: Inter-subnet movement

If the node disconnects from subnet A and connects to subnet B, its IP address will become topologically invalid. Even though the mobile node is connected to a new network, its traffic is still routed to subnet A and subsequently lost. It is possible for the node to use the Dynamic Host Configuration Protocol (DHCP) to receive a new valid IP address whenever it moves to a new IP network. However, all network applications and services will have to be restarted to allow the transport and application layers to use the new IP address. For example, a TCP connection is defined by the source and destination IP addresses along with the source/destination port numbers. If one of these quantities changes, the TCP connection will be broken. In some cases, such as within an 802.11 hot spot, DHCP provides an adequate solution. In this scenario, a user may reboot their system or restart their applications before and after connecting to the hot spot. This behaviour has been termed "portability". The difference between network portability and mobility is that portable systems are easily moved from place to place, but their network interfaces are only actively connected while at a fixed location. Mobile systems, on the other hand, are able to remain active as they move through different locations [67].

The aim of next-generation mobile networks is to extend the current "portability model" to include support for complete mobility. The Mobile IP protocol is currently being developed by the IETF within the Mobile IP working group to support network layer mobility management. Mobile IP allows a mobile node (MN) to migrate through different IP networks, while ensuring that all mobility functionality is transparent to upper layers. This transparency ensures that all upper layer sessions (e.g. TCP) are maintained in spite of any network layer movement that may occur. Mobile IP adapts the standard IP routing mechanism such that a MN's traffic is forwarded to its current position as it moves through different networks.

In the same way as the IP protocol has been developed along two separate branches, Mobile IP has followed a similar pattern. The current version of IP deployed in the Internet is IP version 4 (IPv4). As the Internet's popularity has grown over the last few years, the limited IPv4 network address space is quickly being used up. This problem was one of the main motivations for developing a new version of IP. In 1995, the Internet Engineering Task Force (IETF) published its recommendation for IP version 6 (IPv6). IPv6 supports longer 128 bit addresses as opposed to the 32-bit addresses used in IPv4. In addition to relieving the IP address shortage, IPv6 has incorporated many additional improvements as a result of the years of experience with IPv4 [51].

As the design of IPv6 progresses within the IETF, a corresponding version of Mobile IP (Mobile IPv6) is also being being developed. Just as IPv6 builds upon IPv4, Mobile IPv6 is the successor to Mobile IPv4. In addition, the fact that IPv6 has not been widely deployed has allowed Mobile IPv6 entities to be more strongly integrated into the IPv6 protocol. Mobile IPv6 is functionally similar to Mobile IPv4, but includes many built-in enhancements over the previous version. Because of these factors, most research and development within the field of Mobile IP is being focused on the newer Mobile IPv6 branch.

Mobility functionality can be implemented on different layers of the network protocol stack, ranging from the application layer to the data link layer. Both link layer and network layer mobility have been introduced above. Future IP-based mobile networks will support a combination of link layer and network layer mobility management [13]. An example of such a network (although there are many others) would use 802.11 technology to enable link layer mobility while using Mobile IP to enable network-level mobility management.

The issues related to network layer mobility in general are fundamentally different to those

of other layers. Cellular services are an example that highlight this difference. Modern cellular systems have set a precedent in terms of effective mobility management. In some cases, cellular network coverage encompasses entire geographic regions, resulting in ubiquitous telephony access. These networks are able to cater for large numbers of highly mobile users. Furthermore, many of these cellular networks are moving further and further into the realm of data services. The deployment of 3G and 2.5G networks that support services such as General Packet Radio Service (GPRS) are growing steadily. For example, in Japan, NTT DoCoMo's i-mode system supports mobile data services such as web-browsing, email, and on-line-banking[2]. These cellular data services offer relatively low data rates when compared to wireless LANs. 2.5G GPRS currently offers approximately 100 kbps while next generation 3G systems will offer between 2-4 Mbps. However, their broad coverage areas allow a much higher range in terminal mobility.

Both wireless LANs and cellular data services described above are examples of technologies that implement mobility at the link layer. These examples serve to illustrate that both of these networks support terminal mobility very effectively (within the confines of their technical limitations). Although mobility functionality is very complicated and specialised at lower layers, this results in mobility management that may be highly optimised [13]. In contrast, although Mobile IP offers a simple network layer mechanism to support IP mobility, it suffers from serious performance drawbacks. This is especially true when compared with the mobility offered by the cellular systems described above. Scalability is an example of one of these problems, and the standard Mobile IP protocols (both versions 4 and 6) are unable to cope with the large numbers of highly mobile nodes seen in cellular networks. These technical and practical issues facing Mobile IP are the main factors preventing its wide-spread deployment.

One of the main challenges that the Mobile IP protocol faces is its poor performance during handoff. When a Mobile IP node moves from one IP network to another it executes a network layer handoff. During this handoff, a mobile node must reconfigure its IP address and other IP settings. The forwarding of a MN's traffic must also be adjusted to reflect the MN's new topological position. A Mobile IP handoff can thus lead to performance degradation due to long handoff latencies, where a MN is temporarily disconnected from all IP networks. In this time, packets are misrouted and subsequently lost, introducing significant interruptions in the MN's traffic flow. This obviously has an adverse effect

[2]Further information can be found at `http://www.nttdocomo.com/corebiz/imode/global/index.html`

on upper-layer services and applications. For example, TCP connections are especially affected by such delays because of the congestion control mechanisms that they employ. An exponential back-off is used to lower the frequency at which unacknowledged packets are retransmitted. When handoff delays are long, the retransmission timeout will also become large, causing additional delays [38].

The effects of a Mobile IP handoff depend strongly on the type of application being supported. The interruption that a Mobile IP handoff introduces may not be extremely significant when considering traditional Internet applications such as web browsing and FTP. Users of such applications may notice a slight delay in responsiveness, however the applications would recover soon after handoff is completed. This is because these applications have relatively loose requirements with regards to latency and packet loss. However real-time applications with stricter requirements would perform unacceptably during a Mobile IP handoff. Both handoff latency and packet loss must be reduced significantly if these applications are to be supported.

An example of such a real-time application is Voice over IP (VoIP). VoIP is an IP telephony technology whereby voice information is encoded, compressed, packetized, and transmitted over an IP network (typically the Internet) from one network host to another. The main motivation behind developing VoIP technology is that it allows the convergence of both voice and data services within a unified infrastructure. Therefore, just as cellular networks are increasingly providing for data services, 802.11 wireless LANs are also moving into the realm of voice communications. There has been a great deal of research into the effectiveness of using 802.11 networks to support Voice over IP (VoIP) [76, 53, 44, 72].

When considering the effectiveness of transporting VoIP traffic over a particular network technology, a number of factors need to be taken into account. Some of the key requirements of a VoIP system are that the end-to-end delay, delay variation (jitter) and packet loss must be kept within certain limits. If these restrictions are not adhered to, users will be unable to carry out an intelligible two-way conversation. Thus, within the context of VoIP, one of the challenges facing Mobile IP is ensuring that handoff latency and packet loss fall within the limits imposed by this class of services.

Figure 1.2 extends the wired model previously presented in Figure 1.1 and will be used to introduce the Mobile IP handover process. Figure 1.2 illustrates a wireless network infrastructure that combines both link layer and network layer mobility management, using 802.11 and Mobile IP respectively. The network model below is an example that consists

of two different IP subnets, both connected via a gateway to the Internet. Within each subnet, an 802.11 wireless LAN has been set up. The 802.11 access points (APs) provide the MN with wireless access to the wired infrastructure. The wireless LAN's coverage area has been increased by using more than one access point per subnet, as shown below.

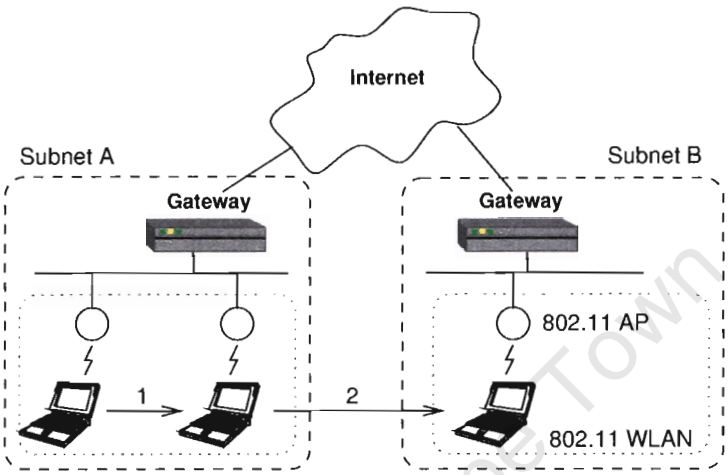

Figure 1.2: Wireless Internet access network model

A mobile node attaches itself to a particular subnet by establishing a wireless link between its 802.11 interface and an 802.11 AP. The MN is then able to move within the geographic bounds of the wireless LAN. As it moves within the wireless LAN, the mobile node's wireless link quality may become degraded (e.g. as the MN moves further away from the AP). The wireless interface may therefore associate itself with a different access point that will provide a better wireless link quality (1). This process of moving from one AP to another is termed link layer handoff. Although the MN may perform many link layer handoffs within a particular network's wireless LAN, these handoffs are transparent to the network layer (IP) because the MN is still connected to the same IP subnet.

On the other hand, if the mobile node migrates to subnet B, it will have to perform both a link layer and a network layer handoff (2). Specifically, the MN's wireless interface will firstly have to establish an 802.11 link to an AP on subnet B. However, in this intermediate state the MN will have an invalid IP configuration. Therefore, once link layer handoff has been completed, a Mobile IP handoff will continue to ensure that the MN's IP settings are correctly configured for subnet B. The total Mobile IP handoff process described here can be divided into a sequence of distinct stages.

One of the main components of a Mobile IP handoff is the movement detection stage. Movement detection is carried out by a mobile node once it has established a (wireless) link to a new IP network. The purpose of movement detection is to allow a MN to become aware that it is connected to a new IP network. As Mobile IP was designed to operate independently from the link/physical layers, movement detection only uses IP mechanisms to detect the arrival of a MN on a new network. As a result, movement detection in its generic form is inefficient and contributes significantly to the total Mobile IP handover delay.

Recently, much research has been devoted to optimising the movement detection process in an effort to reduce Mobile IP handoff latencies. For example, the "Detecting Network Attachment" (DNA) working group was established within the IETF in February 2004. This working group has focused primarily on defining how the IP layer should detect changes in the link layer (for example, as a result of a link layer handoff) along with the best approach for dealing with these changes[3]. The working group has not focused exclusively on the issues relating to movement detection nor on Mobile IP handoff in general. Rather, the main purpose is to define the "best common practice" for IP nodes that wish to detect changes in their link layer state quickly and efficiently. Despite this, many of the drafts and discussions produced by members of the DNA working group are of direct relevance to improving Mobile IP movement detection.

The basis for many improvements to movement detection is that the independence between the Mobile IP layer and the link layer should be reduced. Instead of relying purely on IP mechanisms, Mobile IP should make use of link layer information to improve the detection of movement to a new network. Obviously, the specific format of this information is highly dependent on the type of link layer used, along with the availability and accessibility of link layer parameters. For instance, simple indications of link layer events, called "triggers", can be relayed to Mobile IP. Alternatively, more extensive information, including link layer parameters, can be accessed.

1.2 Thesis Objectives

This study investigates ways of improving Mobile IP handoff performance by optimising movement detection. Movement detection is a significant stage in a Mobile IP handoff and

[3]Some details of the official charter are still being discussed at the time of writing.

therefore improvements to the movement detection process will result in faster Mobile IP handoffs. The specific focus of this study lies in the investigation of different techniques and architectures that will bring about improved movement detection. These techniques aim to minimise the latency and the corresponding packet loss contributed to the total Mobile IP handoff by the movement detection stage. There have been a number of movement detection optimisations proposed in the literature and many of these have been discussed within the DNA working group. The most significant of these proposals will be introduced in later chapters.

This study specifically focuses on the integration of Mobile IP and 802.11 wireless LANs within a single network architecture. This architecture was introduced in Figure 1.2 and will incorporate both link layer and network layer mobility management. Despite the fact that a number of improvements to both movement detection and Mobile IP handoff in general have been suggested, many are not applicable when considering 802.11 networks. Some of the assumptions made by these improvements are invalid when considering 802.11 networks because 802.11 network interfaces indirectly impose certain restrictions on network layer mobility.

Thus, in order to fully understand how the performance of movement detection may be improved, the specific characteristics of 802.11 wireless LANs must be explored. In order to achieve this, the properties of 802.11 link layer handoffs and their effects on the subsequent movement detection will be investigated. Many studies have been carried out that investigate 802.11 link layer handoffs [57, 76]. The aim of this part of the project is to both verify their results and to gauge the effects that these handoffs have on Mobile IP handoffs.

A theoretical study of these movement detection optimisations that highlights their advantages, disadvantages and general implications will be presented. However, in order to quantitatively analyse these different techniques, an evaluation framework will be used. A simple network incorporating the main Mobile IP entities will support the movement of a MN from one logical IP network to another. This framework will facilitate research into mechanisms that enable link layer information to be communicated to Mobile IP. The evaluation framework will be used to design and implement practical optimisations to movement detection. It will also enable a detailed analysis of the extent to which these optimisations improve Mobile IP handoff performance over the generic movement detection mechanisms.

Some of the techniques that are expected to result in the fastest movement detection, also induce unstable behaviour under certain circumstances. For example, a MN may be allowed to perform unnecessary handoffs. The factors that cause this instability will be illustrated. A hybrid mechanism will be presented that incorporates several different techniques with the aim of ensuring that movement detection performs stably.

In this study, VoIP will serve as an example of a real-time application with certain network requirements. One of the reasons for choosing VoIP is that it is the first advanced application that has experienced significant deployment in the Internet [56]. The requirements of a VoIP system will provide a benchmark for evaluating Mobile IP handoffs using different techniques. The delays and packet loss introduced by movement detection optimisations will be compared to VoIP criteria in order to establish if such real-time applications can be supported. In addition, the output voice quality will serve as an indication of how successfully such a system can support VoIP audio traffic.

Lastly, a brief theoretical investigation will be carried out to determine if Mobile IPv6 includes any mechanisms that would improve movement detection performance over Mobile IPv4.

1.3 Scope and Limitations

This study focuses specifically on techniques that improve movement detection performance. Most of these techniques introduce a degree of synchronisation between the Mobile IP and 802.11 layers.

A number of mobility management protocols exist besides Mobile IP, some of which operate on different network layers. An example of an application level technology is the Session Initiation Protocol (SIP). These alternatives to performing mobility management using Mobile IP will not be investigated. Research will be restricted to network and link layer mechanisms.

Mobile IP handoff represents a broad area of research. The numerous ways to improve Mobile IP handoff cannot all be discussed in this study. Some of these improvements will be introduced briefly at a later stage merely to illustrate certain ideas. However, the main area of this investigation has been restricted to movement detection. One of the reasons that Mobile IP handoff is such a broad topic stems from the fact that Mobile IP operates over different technologies. Handoffs therefore take on different properties depending on the

underlying technology. Only horizontal Mobile IP handoffs between homogeneous 802.11 wireless LANs are considered in this study. Vertical handoffs, such as those between 802.11 and GPRS are not considered.

Many of the problems in synchronising 802.11 and Mobile IP arise from the fact that 802.11 functions and parameters have been hard-coded into the hardware and firmware of commercial wireless cards. The amount of 802.11-specific information that is accessible to a mobility system is often dependent on the particular driver or firmware used. This makes the generation of link layer hints device dependent and the examination of link layer handoff procedures extremely difficult. It is also impossible to modify these handoff procedures, which makes their characteristics dependent on the particular device.

The subjective effects of a Mobile IP handoff on the output voice quality of a VoIP application will be used to evaluate different movement detection techniques. Subjective methods will be employed because they are widely applicable to various types of degradation, including "erased frames that occur in systems such as mobile communications)" [49]. The relatively simple tests used in this study have their limitations in that they are somewhat artificial. However, these methods are useful in illustrating well-defined trends in user opinions in spite of their drawbacks. These factors will be discussed further in later chapters.

Although VoIP will be used to evaluate the results of movement detection optimisations, the details involved in establishing and maintaining VoIP calls will be ignored. Furthermore, different audio codecs and the effects that handoffs have on their output will not be investigated. Tests performed using the evaluation framework involve a single mobile node that participates in a single point-to-point VoIP call.

Security is an important issue within the Mobile IP working group. Factors such as this, along with issues relating to Authorisation, Authentication and Accounting (AAA) remain on the periphery of this study.

1.4 Thesis Outline

The remainder of this document is organised as follows:

Chapter 2 provides the background information that will form the foundation of the rest of this study. Initially, the principles of transporting voice information over a data network as

part of a VoIP application are discussed. A survey is performed of the network requirements needed to effectively transport VoIP traffic. Next, the 802.11 and Mobile IP technologies that make up the network architecture that will support this application are introduced. An outline of both the Mobile IPv4 and Mobile IPv6 protocols is presented along with a description of their important differences. This will provide a foundation for several design decisions that arise in later chapters. Special attention is also placed on understanding the 802.11 handoff process and how it is facilitated in wireless LANs.

Chapter 3 subsequently focuses more specifically on the Mobile IP handoff process. Different architectures and techniques that improve Mobile IP handoff are discussed relative to the network model presented in Figure 1.2. The movement detection process is then introduced and the differences between the Mobile IPv4 and Mobile IPv6 mechanisms are highlighted. Optimisations discussed in the literature and within the DNA group are also explained. Lastly, a hybrid mechanism that combines different optimisations is proposed.

Chapter 4 includes information on the design and software architecture of the different movement detection optimisation schemes that will be implemented on the evaluation framework. The practical implications of these schemes, along with different implementation techniques, are also discussed here. This section focuses specifically on the software designs that will support the communication of 802.11 hints to the Mobile IP layer.

In Chapter 5, an overview of the evaluation network is presented, including a description of the Mobile IP implementation. The network configuration that will provide a platform for testing movement detection performance is described. Each entity present in the network is explained, along with details on how they were designed and developed. This includes information on both the hardware and software systems used. Furthermore, the latency, packet loss and VoIP quality assessment experiments performed on the evaluation framework are described. Finally, minor modifications to the evaluation framework were required in order to fully test the hybrid system. These are illustrated at the end of this chapter.

Chapter 6 presents results from tests performed on the evaluation framework, together with an analysis of these results. Different movement detection techniques are compared using handoff latency and packet loss as metrics. A description of the method used to assess the quality of a VoIP call during a Mobile IP handoff is subsequently outlined. The results of these assessment methods are also used to compare different movement detection mechanisms. Lastly, an analysis of the hybrid system's performance is presented.

Chapter 7 presents a set of conclusions that were drawn from these evaluations. This chapter also contains concluding remarks on several issues raised in previous chapters.

Chapter 8 lists some recommendations that were noted during the course of this project. They may be used to prescribe future work that further develops the evaluation framework. Areas of research related to this study that promise to be active in the near future are also described.

Chapter 2

Background Theory

2.1 Introduction

The previous chapter briefly introduced several different network technologies including VoIP, Mobile IP and 802.11 wireless LANs. However, before an in-depth discussion of these systems can take place, it is important to understand the broader paradigm that defines how hosts communicate information over a network.

Network architectures and systems usually incorporate a number of different interworking technologies, such as those mentioned above, and therefore have a highly complicated structure. In order to manage this complexity, network models group functions that allow a host to communicate over a network into a number of logical layers. This layered structure simplifies the design of individual network technologies.

Both Mobile IP and 802.11 offer mobility management mechanisms that support end-user applications. Although they are all interlinked, these technologies operate on different planes within a mobile network node. This is illustrated using the Open Systems Interconnect (OSI) network model (Figure 2.1). The OSI model describes a hierarchical networking protocol stack where the details of a given layer (e.g. specific technology or protocol) are abstracted to upper layers. This architecture allows the complexities associated with transmitting information over a network to be localised at specific layers. Each layer offers certain services to higher layers, while hiding the intricacies of how those services are implemented.

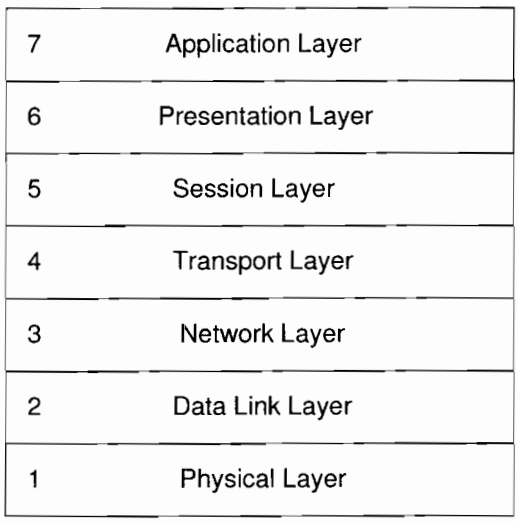

Figure 2.1: The OSI Network Model

Each layer in the OSI model fulfils a specific function and purpose. A network protocol residing within a given layer is therefore independent from both its higher and lower-layer protocols. This allows implementations of specific layers to be changed without having to replace the entire protocol stack. The following example, given within the context of this study, provides an illustration.

802.11 is a data link layer (layer 2) technology used to establish wireless connectivity over relatively short distances. Mobile IP focuses on larger scale movement between IP networks, functioning at the network level (layer 3). Because they reside on different layers of the networking stack, the mechanisms used by both these technologies operate in isolation. This has traditionally been an important factor in the design of networking protocols. For example, layer independence allows Mobile IP to operate over heterogeneous link layer technologies such as 802.11 wireless LANs, GPRS or Ethernet. However, as was previously introduced, this independence is the very aspect that results in poor Mobile IP handoff performance.

This chapter is devoted to providing background information that will be built upon in subsequent chapters. As the relevant technologies are discussed, it is important to keep them in perspective, relative to the model described above.

The focus of this project lies in the performance of Mobile IP handoff in 802.11 networks

and how it may be improved. When developing architectures involving these technologies, it is important to consider the types of applications that will be supported by the network. Voice over IP (VoIP) therefore serves as an example application that will ascertain if this type of real-time interactive service can operate on an access network that combines Mobile IP and 802.11. Although VoIP is not central to this study, it is introduced first because its requirements serve as goals for the system architecture that is detailed in this chapter. Therefore, the mechanisms used to transport voice information over a data network as part of a Voice over IP application will be introduced in the following section. The network requirements of VoIP will also be outlined.

The discussion will subsequently move further down the networking model to the network and link layers. A brief overview of both Mobile IP and 802.11 technologies will be provided. In order to understand the interaction between Mobile IP and 802.11 during a Mobile IP handoff, these technologies must first be studied individually. It is also important to delve into some of the details of the 802.11 standard in order to understand what information is available to the movement detection techniques discussed in later chapters. These details will become relevant when discussing the issues and implications related to both general handoff and movement detection optimisations.

2.2 VoIP

Interactive voice still remains the dominant mode of human communication. This form of communication has been traditionally carried over circuit-switched networks such as Public Switched Telephone Networks (PSTN). In these networks, resources such as bandwidth are reserved when a dedicated connection (or path) is established. However, as the bandwidth capacities of data networks increase, systems that allow voice information to be transported over existing data networks are becoming economically attractive. Because of this, VoIP systems are being increasingly deployed in packet-switched IP networks such as corporate intranets [56]. This section presents an overview of VoIP along with the requirements that these applications impose on the underlying networks.

2.2.1 Overview

A VoIP system (shown in Figure 2.2 below) can be subdivided into four components: signalling, encoding, transportation and gateway control [55]. A signalling protocol such

as SIP or H.323 is used to set up and manage a VoIP call between the end hosts. An audio codec transforms the input analogue voice signal into an appropriate digital format called a frame. The digitised voice frame is then packetized and transmitted over an IP network. The network is responsible for transporting voice information through the network without affecting the voice or conversation quality. At the receiver, incoming packets are converted back into an analogue output signal. Lastly, gateways allow VoIP systems to interoperate with different networks such as a PSTN. This section will only discuss the relevant transportation aspects of a VoIP system.

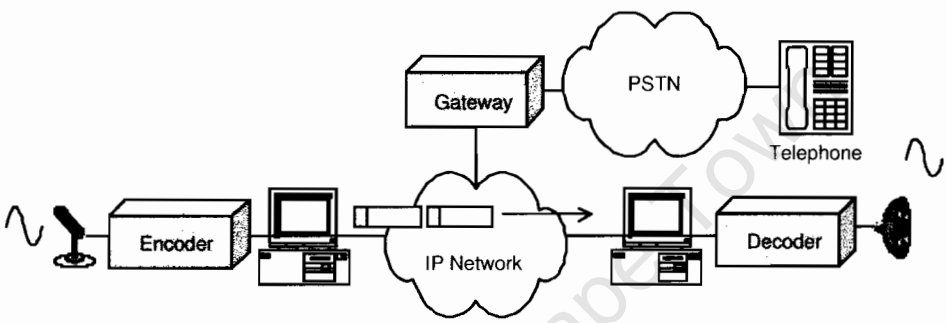

Figure 2.2: Basic VoIP system overview

Several issues arise when transmitting voice information over best-effort connectionless networks, due to the fact that voice is a sensitive real-time application. For example, packets can experience long and variable delays in such networks. Packets are also not guaranteed to reach their destination. All of these factors affect the voice quality received at the end hosts. The next section defines some of the requirements to which an IP network must conform in order to support VoIP.

2.2.2 Interactive Voice Requirements

There are three main criteria that must be catered for when transmitting voice over a data network. Due to its interactive nature, voice information must reach remote hosts within a certain time limit. VoIP therefore requires the underlying network to place strict bounds on end-to-end delays to which packets may be subjected. Another important factor that must be limited is delay variation (jitter). Jitter defines the short term fluctuations in packet arrival rates. Lastly, VoIP traffic requires low packet loss in order to sustain

acceptable output audio quality. The descriptions listed below describe the effects that these parameters have on output voice quality.

Delay

The following table outlines the effects that different delay values have on the perceived voice quality [56, 50].

One way delay	Effect
< 100-150 ms	Delay not detectable
150-250 ms	Acceptable quality, but slight noticeable delay/hesitation
Over 250-300 ms	Unacceptable delay

Table 2.1: Delay guidelines for VoIP

The end-to-end delay between two applications is essentially the sum of the delays that traffic experiences at every stage in the communication path. This includes the delays from processes such as encoding, packetization, network transmission and buffering. From Table 2.1 it is clear that the end-to-end delay between two VoIP applications should ideally be kept below 150 ms. Between 150 ms and about 250 ms the effects of delay are noticeable, however the output voice quality is still satisfactory to the majority of users. The effect of delays in this range has been characterised as "a slight hesitation in the response of the conversational partner" [50]. When the delay exceeds 300 ms, conversation becomes almost impossible[1].

Jitter

The following table outlines the effects that different jitter values have on the perceived voice quality [56, 50].

Packet delay variation should be kept below 40 ms to ensure correct play-out at the receiver. Table 2.2 illustrates that jitter of up to about 70 ms is acceptable. Jitter can be alleviated through the use of a de-jitter buffer at the receiver end-system. The size of a de-jitter buffer is an important consideration as it adds to the total end-to-end delay.

[1]These figures assume echo cancellation has been performed. Without echo cancellation, the end-to-end delay requirements would be reduced to about 30 ms [56].

Delay variation	Effect
< 40 ms	Jitter not detectable
40-70 ms	Acceptable quality, but slight noticeable delay/jumble
Over 75 ms	Unacceptable jitter

Table 2.2: Jitter guidelines for VoIP

Packet loss

The previous two parameters affect the conversational quality of a VoIP connection. In contrast, packet loss affects mainly the quality of the output audio delivered to the end user. Unfortunately, the maximum limitations for packet loss imposed by a VoIP system, along with the resultant effects, are not as clearly defined as for the previous two phenomena. The reason for this is that the effects that packet loss have on the output voice quality are dependent on many factors, such as the properties of the particular audio codec and the error correction or concealment mechanism used.

The pattern of packet loss is also a significant factor in evaluating the degradation of output voice quality. For instance, a VoIP codec may be able to deal gracefully with individual packet losses that occur randomly during a conversation by using an error concealment mechanism at the receiver. However, the loss of a contiguous segment of traffic can cause a significant degradation in the voice quality. Furthermore, if this "bursty" loss occurs during a period of silence, the effects will be negligible as compared to a period of speech. The most significant degradation occurs when packet loss coincides with the beginning of a voiced segment. This is because most error concealment techniques attempt to mask lost frames using information from a previous frame, which in this case is unvoiced [73]. In addition to this, certain advanced codecs produce output packets that may be dependent on previous packets. As a result, "bursty" packet loss can induce a period of audio distortion that may extend significantly longer than the period of packet loss. Advanced codecs are therefore more sensitive to packet loss [56].

Packet loss can be caused by a number of factors. For example, voice packets may be dropped or lost in the transmission network. Packet loss can also stem from packets that arrive at the end system, but have exceeded the delay or jitter requirements.

2.3 Mobile IP Protocol

In current IP networks, a node's IP address uniquely defines its network point of attachment. A node must therefore be located on the network indicated by its IP address in order to receive packets addressed to it. Mobile IP provides a relatively scalable mechanism that allows a mobile node to continue its IP communications as it migrates through different IP subnets. This is achieved without changing its primary IP address, thereby maintaining IP connectivity along with any upper-layer sessions.

Mobile IP was designed to operate transparently alongside existing networking layers and entities. This means that a mobile node does not need to use a special mobility-enhanced protocol stack, nor run specific mobility-aware applications. Furthermore, Mobile IP was designed to function compatibly with existing IP end-systems and routers.

As stated earlier, the Mobile IP protocol has been developed within two different branches. Mobile IPv4 operates within the current IPv4 framework while Mobile IPv6 was designed within the context of IPv6. This section is devoted to investigating these two different versions of the Mobile IP protocol. To begin with, a general overview of the Mobile IP protocol will be given within the context of Mobile IPv4. The Mobile IPv6 protocol will then be introduced by describing some of the enhancements that it has incorporated.

2.3.1 Mobile IPv4

Mobile IP defines two types of networks, a home and foreign network. A *home network* can be loosely defined as the network where a mobile node spends most of its time. This definition, while not completely accurate, will suffice for the purposes of this study. A *foreign network* is any other IP network that will support visiting mobile nodes. The Mobile IP protocol also defines the following entities:

Mobile node A mobile node (MN) is a network host that changes its point of attachment from one IP network to another. It maintains its communication with remote hosts even after it has connected to a different network. The MN usually connects to an IP network using a link layer technology such as Ethernet or 802.11.

Home agent The home agent (HA) is a router on the MN's home network that forwards the MN's traffic to its current point of attachment when the MN is away from the home network.

Foreign agent A foreign agent (FA) is a router on a visited network that cooperates with a home agent to deliver traffic to a visiting MN. Home and foreign agents are collectively called mobility agents.

The central element that supports IP mobility is the allocation of more than one IP address to a mobile node. A MN is assigned a primary IP address called a home address. The home address, like conventional IP addresses, is a relatively permanent address. It essentially represents the MN's identity on the network. While a MN resides on its home network, all mobility services and entities are inactive and the MN uses only its home address to communicate. For example, the home address is used to define TCP connections established between the MN and corresponding nodes. When a MN moves to a foreign network, it is allocated a second globally-routable IP address, called a *care-of address* (CoA), which is valid on the visited network. The CoA represents the MN's current network point of attachment and reflects its topological position in the foreign network. The CoA is thus a transient address, changing as the MN migrates through different foreign networks.

Mobile IP can be divided into three subsystems. These are mobile agent discovery, registration and tunnelling. Agent discovery is the process whereby a MN detects the presence and attributes of a new mobility agent. The mechanisms used to achieve this are similar to the ones used by conventional network nodes to detect IP routers. A mobility agent uses periodic advertisements to announce its presence to all listening MNs. Agent discovery usually occurs when a MN migrates to a new IP network. A MN will use these agent advertisements to discover new mobility agents on the network. Agent discovery mechanisms will be explored in greater detail in the next chapter.

When a MN is visiting a foreign network, the home agent and the foreign agent cooperate to deliver IP packets addressed to the mobile node. However, a mobile node must first configure and register its CoA before it attempts to send or receive IP packets on the visited network. A mobile node can obtain a CoA from a foreign network using one of two modes. In the first mode, the MN receives a *foreign agent care-of address* directly from the foreign agent. The foreign agent supplies the MN with its own IP address to use as a CoA. In the second mode, a MN relies on an external mechanism such as DHCP to receive a local IP address. This type of address is called a *co-located CoA* and is independent from the foreign agent's IP address.

The home agent maintains a mapping between a MN's home address and CoA called a binding. A particular binding also has a lifetime associated with it to ensure that if a MN

does not renew its binding, the binding will expire and be deleted. The binding entries and corresponding lifetimes of several MNs are stored in a binding cache. A MN's binding must be updated as the MN moves through different IP networks to ensure that packets are forwarded to the correct foreign network. Therefore, once a MN has acquired a new CoA, it notifies its home agent that it is reachable through a new address. The registration process allows a MN to relay its new CoA back to its home agent and update its CoA binding.

Once registration has been completed, the home agent must also intercept packets destined for the MN. In order to do this, the home agent uses the proxy and gratuitous ARP (Address Resolution Protocol) mechanisms. A gratuitous ARP broadcast is used to update the ARP caches of all local network nodes when a MN first moves away from the home network. Proxy ARP is used thereafter to allow the home agent to respond to ARP requests on behalf of the MN. When a host broadcasts an ARP request for the MN's MAC address, the home agent includes its own MAC address in the ARP reply. This ensures that all nodes residing on the MN's home network (including switches and routers) forward the MN's traffic to the home agent.

Figure 2.3 and the description that follows illustrate how a mobile node is able to receive IP traffic while connected to a foreign network.

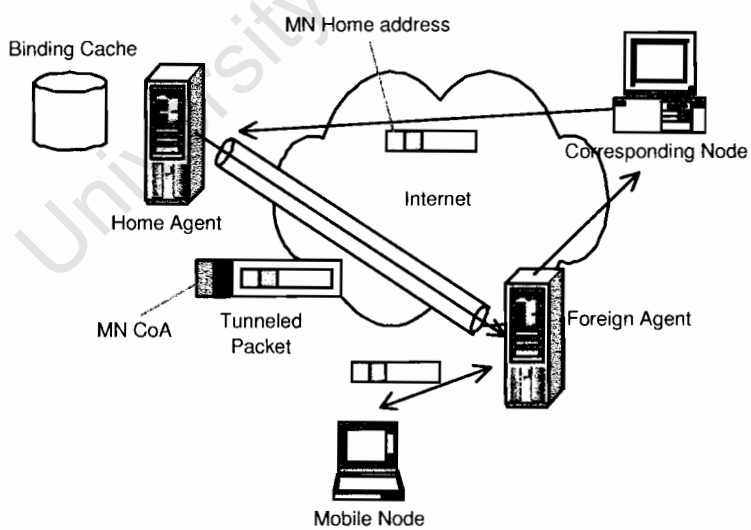

Figure 2.3: Mobile IP tunnelling (foreign agent CoA mode)

A corresponding node uses the home address to send packets to the MN. Any IP packets addressed to the mobile node's home address are routed to the home network using standard IP routing mechanisms. While the MN is away, the home agent intercepts IP packets destined for the MN's home address on its behalf. These packets are then tunnelled by the home agent to the MN's current point of attachment using the CoA. The home agent first uses its binding cache to look up the MN's current CoA. Tunnelling is then usually achieved by encapsulating the original packet in a new IP packet with the MN's CoA as destination address (IP-in-IP encapsulation)[2]. This is done in order to shield the original inner packet from the intermediate routers between the home network and visited network (see Figure 2.4). The tunnelled packet is then routed to the foreign network, again using normal IP routing. In foreign agent CoA mode, a foreign agent receives the tunnelled packet, extracts the original packet and delivers it to the mobile node. In the co-located CoA mode, the MN receives tunnelled packets, and decapsulates them itself. Because packets leave the tunnel unmodified, applications running on both the MN and corresponding nodes do not have to be mobility-aware. A CoA can therefore be thought of as defining the endpoint of a tunnel where the original packet emerges, as seen in Figure 2.4. In both addressing modes, packets sent by the mobile node to a corresponding node are either routed directly or tunnelled back through the home agent.

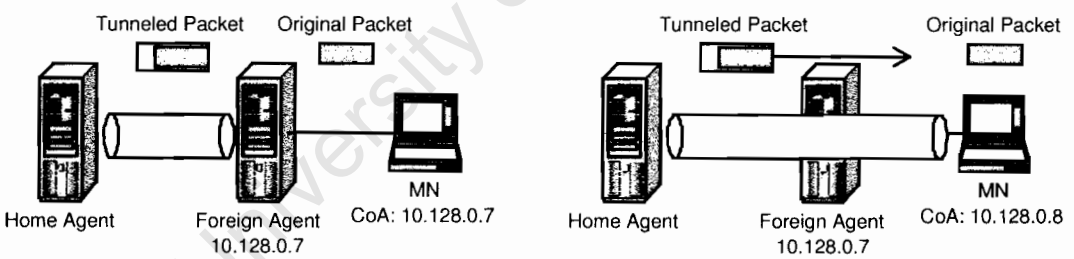

Figure 2.4: Foreign agent CoA (a) Co-located CoA (b)

It is important to note that in order for the foreign agent CoA address scheme to function correctly, the MN must be on the same link as the foreign agent. This is because the foreign agent delivers the original decapsulated packet to the MN over a link layer protocol using the MN's link layer (MAC) address. The decapsulated packet's IP destination address can no longer be used because the MN is not on its home network.

[2]Alternative tunnelling mechanisms are specified in Mobile IPv4 RFC 3344 [28].

Mobility agents have a susceptibility to becoming bottlenecks when considering large numbers of MNs. This is because all of a MN's traffic flows through these agents while it is away from home. One way of protecting against this is by deploying several home and foreign agents on a particular network to share the load. Several home agents can be managed using the automatic home agent discovery [28]. A MN can send a registration message to a broadcast address which is used to discover all available home agents. Home agents that are at full capacity can select not to reply to this message. Therefore, while a MN is away from its home network, it can use any one of these available home agents, depending on their congestion levels. For more information on this mechanism, refer to the Mobile IP specification [28]. The implications of deploying several foreign agents in a similar fashion will be explored in a later chapter.

2.3.2 Mobile IPv6

The IPv6 protocol [23] is intended to replace the current IPv4 implementations in use throughout the Internet. Mobile IPv6 extends Mobile IPv4, making use of new IPv6 mechanisms that support mobility[3]. In addition, Mobile IPv6 allows these mobility entities to be better integrated into the IPv6 protocol. For example, where as new mobility agents had to be defined in Mobile IPv4, these functions have been assimilated to a large extent into IPv6. Relevant aspects of the IPv6 protocol that have been changed from IPv4 will be presented below along with how Mobile IPv6 makes use of these new mechanisms. The issues highlighted below will be relevant in the following chapter when movement detection techniques are discussed.

IPv6 Addresses

The most obvious difference between IPv4 and IPv6 is that IPv6 addresses are 128 bits long while IPv4 addresses are only 32 bits long. A section of the IPv6 address space has been reserved for IPv4 addresses to facilitate the interworking of IPv4 and IPv6 networks. Like IPv4, IPv6 defines several types of IP addresses including loopback and global unicast addresses. However, in addition to a globally-routable unicast IPv6 address, the network interfaces of all IPv6 nodes are assigned a link-local IP address. A link-local address is

[3]In June 2004, Mobile IPv6 evolved from its Internet-Draft status and was published in RFC 3775 [51] as a proposed standard by the Mobile IPv6 working group.

guaranteed to be unique on a given link (link-only scope) and may be used to communicate with neighbouring hosts on the link. These addresses are not used when packets are to be routed to other networks. Instead, link-local addresses are usually used to communicate with local routers. IPv6 routers send and receive Neighbour Discovery messages (section 2.3.2) using their link-local addresses. A host also uses its link-local address to configure a new global address [62].

IPv6 Address Autoconfiguration

A Mobile IPv6 node is assigned two separate IP addresses while it is on the home network and acquires a third when it is away from home. The first two addresses are the link-local and home IPv6 addresses. Like MIPv4, a MN is always addressable by its home address. The third address is the CoA assigned to the MN while it is visiting a foreign network. Both the CoA and the link-local address change as the MN moves through different networks because they are valid only on a particular link/network. An MN can automatically configure both its CoA and link-local address by using either stateful or stateless address configuration mechanisms.

An IPv6 node performs stateful address configuration by querying an external server. The server maintains a database of available addresses along with a list of nodes that have already been assigned addresses. An example of a stateful configuration mechanism is DHCPv6. On the other hand, stateless address autoconfiguration allows an IPv6 host to automatically generate a global IPv6 address without the use of external servers. In order to do this, the node first automatically configures its link-local IP address by combining its network interface's link layer address and the well-known link-local prefix[4]. Once the node assigns a link-local address to its network interface, IP-connectivity with other nodes on the link is established. After validating the link-local address, the node can then autoconfigure a global address using network prefix information advertised by on-link routers and its link identifier (usually the link layer address) [62].

Before either a link-local or a global IPv6 address is assigned to an interface, the IPv6 Address Autoconfiguration draft [62] specifies that a host must verify that the address is unique and not in use by another host. This is achieved by performing Duplicate Address Detection (DAD). A host transmits a broadcast message enquiring if any other nodes are using the address in question. If a reply is received, then the address is already in use

[4]FE80::0

and the autoconfiguration will stop. If after sending several broadcasts[5] no neighbouring nodes have replied, the host can fully assume the IP address [62]. These broadcasts and their replies are called neighbour solicitations and neighbour advertisements respectively, and are part of the Neighbour Discovery protocol introduced below.

IPv6 Header Extensions

Mobile IPv6 tunnelling and registration functions operate similarly to Mobile IPv4. However, IPv6 defines several new extension headers that may be prepended to an IPv6 packet. Both the Destination Options and Routing headers are relevant to IPv6 mobility management. The Destination Options header allows certain options to be included in a packet that will only be processed by the destination host. The Routing header allows a sending node to specify a router that must process the packet en route to its destination. This is typically used when a packet must be delivered to a destination in a way that differs from the standard IP routing mechanisms. A description of how these headers are used in Mobile IPv6 will be given using the scenario shown in Figure 2.5.

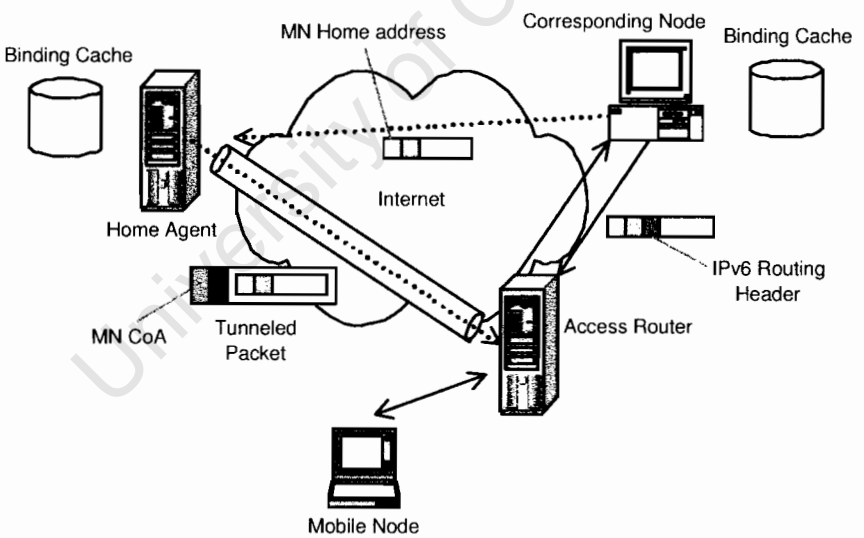

Figure 2.5: Mobile IPv6 system overview

As in Mobile IPv4, packets destined for a mobile node's home address are routed to the

[5]The exact number is configurable but most documents assume 3.

MN's home network. An IPv6 router functioning as a home agent intercepts these packets and tunnels them to the current CoA using IPv6-in-IPv6 packet encapsulation. However, like the home agent, IPv6 corresponding nodes are also able to maintain local bindings for the MNs with which they are communicating. They are therefore able to avoid the routing indirection caused by tunnelling (see Figure 2.3) because packets are sent directly to the MN's CoA, bypassing the home network. These packets are not tunnelled using IP-in-IP encapsulation like the packets forwarded by the home agent. Instead, a corresponding node addresses packets destined for the MN using the MN's CoA. The Routing header extension carries the MN's home address which is used by the MN as the final destination address for the packet. When a mobile node sends a packet, the CoA is used as the source address and its home address is included in a Home Address Destination option. The alleviation of this routing asymmetry has been termed *route optimisation* [51].

A MN registers its current CoA with both its home agent and corresponding nodes. These registration messages do not have to be sent as individual packets. Binding updates are sent as a new type of IPv6 Destination options header that may be included in any IPv6 packet.

Neighbour Discovery

As was previously mentioned, there is no need to deploy specialised foreign agents in Mobile IPv6. Their functionality can be accomplished by IPv6 routers, called *access routers* (ARs). Access routers use integrated IPv6 protocols such as Neighbour Discovery and Address Autoconfiguration to support visiting mobile nodes. Neighbour Discovery [61] defines a set of IPv6 mechanisms whereby a node can discover information about other nodes on the same link. These mechanisms allow a neighbouring node's presence and link layer address, along with information about local routers, to be determined. A node is also able to confirm whether a neighbour is still reachable using Neighbour Discovery [61].

A Mobile IPv6 node uses Neighbour Discovery (similar to Mobile IPv4 Agent Discovery) to discover new access routers and refresh links to existing ones. A MN detects an access router's presence by listening for periodic router advertisements. A MN will configure its CoA address and default router settings based on the router advertisements it receives. A MN can also confirm that an access router is bi-directionally reachable using Neighbour Unreachability Detection (NUD). This is especially important in wireless networks where the up-link and down-link communication properties may differ. This may result in situa-

tions where a MN can receive advertisements from an access router but the router cannot receive transmissions from the MN. NUD is mainly used to detect when a MN has moved to a new link and should discover a new access router. These mechanisms are used when a MN performs movement detection and will thus be discussed further in the next chapter.

2.4 IEEE 802.11 Standard

In the early days of wireless LANs, several companies released various incompatible wireless LAN devices[6]. Then in the mid-1990s, in an effort to unify these various protocols, the IEEE published the 802.11 wireless networking standard [46]. In addition, the predominant LAN technology at the time (as it is today) was 802.3, commonly known as Ethernet. The 802.11 standard was therefore made compatible with 802.3, which is why 802.11 is often nicknamed "Wireless Ethernet". A node's 802.11 network interface is referred to as a *station* (STA) by the IEEE 802.11 standards and this naming convention will be adopted throughout this section.

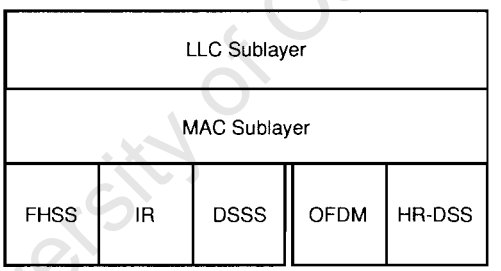

Figure 2.6: 802.11 protocol stack

Figure 2.6 provides an overview of the structure of the 802.11 standard. Like the 802.3 standard, 802.11 focuses on the bottom two layers of the network protocol stack: the data link and physical layers. 802.11, like all other 802 protocols, also divides the data link layer into the 802.2 Logical Link Control (LLC) and Medium Access Control (MAC) sublayers. The 802.2 LLC ensures that all 802 protocols use the same format for the network layer interface, thereby allowing the 802.11 data link layer to provide wired-Ethernet services to upper layers.

[6]such as HomeRF and OpenAir

In practice, a number of intrinsic differences exist between wired and wireless systems. Wireless LANs have dynamic topologies, communicate over a less reliable medium, and all stations may not have full connectivity. The 802.11 MAC sublayer was designed to ensure systematic and fair access to the wireless medium. The MAC sublayer is also responsible for reliable data delivery and hiding the details of the wireless physical layer from the upper layers. The IEEE specification defines two different modes of operation within the MAC sublayer. The first mode is called Distributed Coordination Function (DCF). DCF is intended for asynchronous data transport where all terminals on a link have an equal chance to transmit. DCF uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol to regulate access to the shared medium. This is similar to the CSMA/CD (CSMA with Collision Detection) used by wired Ethernet LANs. The main difference is that reliable collision detection cannot be achieved in wireless mediums and so collision avoidance systems are used. These systems aim to reduce the probability of collisions occurring during transmission[7]. One way that collision avoidance is achieved is through the use of medium reservation messages or virtual channel sensing. When a station has data to send over the wireless medium, it first transmits a Request to Send (RTS) message to the destination station. If the medium is clear, the recipient replies with a Clear to Send (CTS), allowing the sender to transmit over the medium. Information included in these RTS/CTS messages allows all other stations listening on the same channel to estimate how long the medium will be busy. Lastly, the receiving station replies to the sender with an acknowledgement when data has been received correctly. Another collision avoidance mechanism uses physical channel sensing. Before transmission, a station senses the medium and transmits if it is free. If no acknowledgement is received within a specific time after transmission, the station assumes that a collision occurred at the recipient and schedules a retransmission.

The Point Coordination Function (PCF) is the second mode of operation and is designed for time-bound services. In PCF mode, a controlling entity (point coordinator) polls other stations, allowing them to transmit. An 802.11 access point functions as the point coordinator for its connected stations. A station may only transmit a single frame when polled by the point coordinator. Because each station is periodically given the opportunity to transmit, PCF can provide a station with a certain portion of the bandwidth. PCF was designed with the aim of supporting time-bounded applications (to a limited extent), thus

[7]The reason for this is that, usually, radio systems cannot transmit and detect a collision at the same time [33].

providing a certain level of quality of service (QoS). See Appendix B for further details.

In 1997 when the original 802.11 specification was released, three physical layers were defined. The first two are spread spectrum radio techniques, Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS). Both of these radio specifications operate in the 2.4 GHz ISM (Industrial, Scientific and Medical) band. The third physical layer is diffuse infrared. These techniques support data rates of 1-2 Mbps. Other physical implementations have since been added to the base specification in order to support higher data rates and larger ranges. In 1999, 802.11a and 802.11b were released. These standards incorporated two new radio technologies, Orthogonal Frequency Division Multiplexing (OFDM) and High Rate Direct Sequence Spread Spectrum (HR-DSSS) respectively. 802.11a operates in the 5 GHz ISM band and supports data rates up to 54 Mbps while 802.11b operates at 11 Mbps at 2.4 GHz. In 2001, the 802.11g standard was released which uses OFDM technology but operates in the 2.4 GHz band. The maximum bandwidth supported by 802.11g is 54 Mbps.

The 802.11a/b/g standards [47, 45] essentially incorporate new physical implementations into the original IEEE 802.11 specification (along with other minor changes). Therefore, throughout this document, the term "802.11" will refer to the whole family of 802.11 standards, including the newer 802.11a/b/g specifications.

An important feature of the 802.11 standard family is that it support physical technologies that communicate at multiple data rates. This feature is especially applicable to the newer 802.11a/b/g implementations as it allows a wireless connection to degrade gracefully. For example, the 802.11b physical layer is able to support 1, 2, 5.5, and 11 Mbps data rates. Most devices will usually transfer data at the maximum rate (in this case 11 Mbps). However when the link quality deteriorates, lower rates are used to counterbalance poor channel conditions and minimise bit-error rates. Figure 2.7 illustrates the relationship between data throughput and range of these four transmission rates [13].

As two communicating 802.11b stations move further away from each other, they will select lower transmission rates to ensure that data is transmitted reliably. This dynamic rate scaling behaviour has significant consequences within a wireless access network. Firstly, the capacity of a wireless link is greatly diminished when lower rates are used, which may disturb applications with strict bandwidth requirements. Furthermore, the selection of lower data rates negatively affects Mobile IP handoff performance. These issues will be discussed further in later chapters.

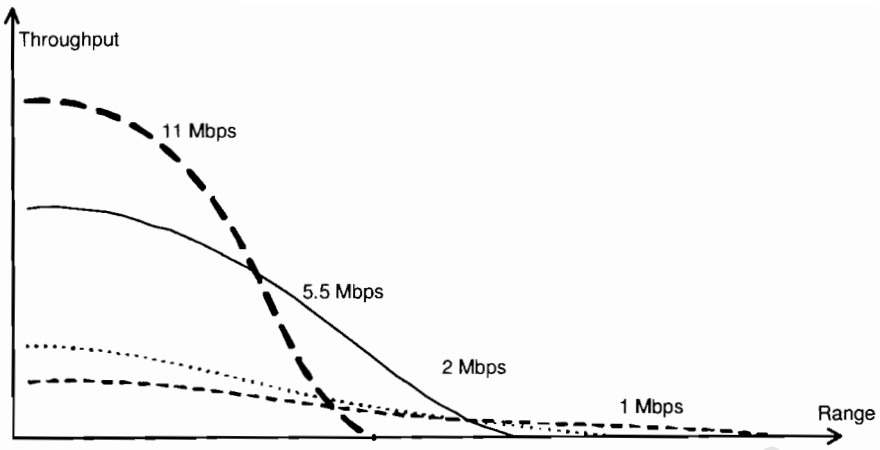

Figure 2.7: Throughput vs. range of different 802.11b modes

2.4.1 802.11 Architecture

The fundamental unit of a wireless LAN defined by the 802.11 standard is the Basic Service Set (BSS). A BSS is a collection of communicating stations within a specific localised area. A BSS is assigned an identifier, called a BSSID, that all participating stations share. The 802.11 standard specifies two modes for how these BSSs can be arranged. The first is ad-hoc mode, where each station can communicate directly with its neighbours. In ad-hoc mode, the set of communicating stations forms an Independent BSS (IBSS). An IBSS (Figure 2.8) is usually formed without any network planning or formal structure and is therefore highly dynamic.

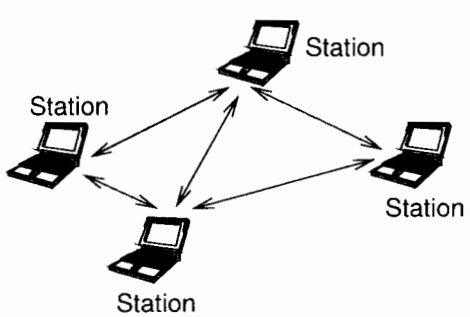

Figure 2.8: Independent BSS (Ad-hoc mode)

Two or more BSSs can alternatively be interconnected in infrastructure mode, where all stations connect to a central 802.11 *access point* (AP) that coordinates their communication. In this configuration, the individual BSSs are grouped to form an Extended Service Set (ESS). An ESS network allows larger and more complicated wireless networks to be formed. An ESS is also assigned an ESS identifier (ESSID) which distinguishes it from other ESSs. Figure 2.9 illustrates the layout of an infrastructure network.

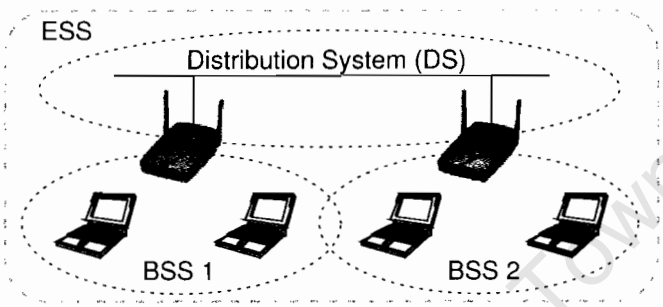

Figure 2.9: Extended Service Set (Infrastructure mode)

Infrastructure wireless networks are more relevant to this study than ad-hoc networks because they are more applicable to the network architecture presented in Chapter 1. For this reason, all 802.11 concepts introduced in this section will be discussed within the context of infrastructure networks.

A Distribution System (DS) facilitates the interconnection and integration of several BSSs to form an ESS. The DS allows individual stations to move between different BSSs transparently to the logical link control layer (LLC). The DS supports the transportation of MAC-level data, management and control messages between the constituent BSSs. The DS medium (DSM) is not specified by the 802.11 standard. The DSM is completely independent from the wireless physical technology, and in practice Ethernet is often used.

The access point is responsible for bridging data between the DS and the wireless stations in its BSS. An AP also incorporates a wireless station entity that allows it to communicate over the wireless medium. An access point will periodically transmit beacon messages to identify its BSS. These beacons also include parameters such as physical layer information and the ESSID. Figure 2.9 illustrates how these elements are used in conventional wireless LANs.

2.4.2 802.11 Services

According to the 802.11 standard, an 802.11 wireless LAN must provide nine services. These services are divided into two groups: distribution services and station services. The five distribution services listed below essentially deal with how stations join and leave a particular BSS. They allow a station to roam within an ESS.

Distribution This service is invoked whenever a station transmits or receives data through an ESS. An AP will receive data traffic from its connected stations through its station interface and will place it on to the DS. It is the responsibility of the distribution service to ensure this data traffic is delivered to recipients within the DS.

Integration When a message has to travel through a non-802.11 network, such as an Ethernet LAN, the message's format must be translated appropriately. The integration service handles this translation.

Association An 802.11 access point manages all the traffic for its registered stations. The association service allows a station to roam transparently between different APs within an ESS. In essence, the association service allows a station to establish a logical connection with a specific access point. Before a station is able to receive data, it must be associated with an AP. This ensures that the DS can correctly deliver the station's traffic to its AP. The same applies when a station wishes to transmit data. If the destination of a data message is a station within the AP's BSS, the AP forwards the message directly to the destination station. If the destination is in another BSS, the AP passes the frame to the DS, which delivers the frame to the destination BSS. Because of these factors, the 802.11 standard specifies that a station may only be associated with a single access point at a time. This is done to ensure that only one AP handles the station's traffic, avoiding potential confusion. However, an AP may support several associated stations.

Reassociation This service is needed when a station moves from one AP to another. It notifies the DS that the station's traffic should be delivered via the new AP.

Disassociation The disassociation service ends an existing association with an AP. It is invoked when a station leaves an AP, either when moving to a new AP or when shutting down.

The next four station services deal with mechanisms used while the station is connected to a BSS (associated with an AP). They provide the same functionality as wired-LANs. Most of these services deal with security issues that are raised by the vulnerability of wireless communications.

Authentication Unlike a wired network, a wireless network does not have precise physical boundaries, which can promote unauthorised access. An 802.11 wireless LAN performs access control using the authentication service. This allows the identity of a connecting station to be challenged and verified. Authentication is performed on the link level, and aims to incorporate the properties of wired networks into wireless LANs. The 802.11 standard provides the following two levels of authentication:

1. Open System
   A station wishing to connect to an AP first authenticates itself by sending an authentication request. In an Open System, the AP immediately replies to an authentication request with a positive authentication response (Figure 2.10).

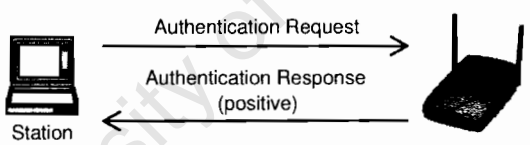

Figure 2.10: Open System authentication

2. Shared Key
   This authentication mechanism uses the wired equivalent privacy (WEP). Access is granted to stations that present a shared, secret WEP encryption key. The message sequence in shared key authentication is shown in Figure 2.11. A connecting station will receive a challenge string from the AP in response to its association request. A station demonstrates its knowledge of the shared key by encrypting the challenge string with this shared key and sending it back to the AP. If the result is correct, the AP replies with a positive authentication response.

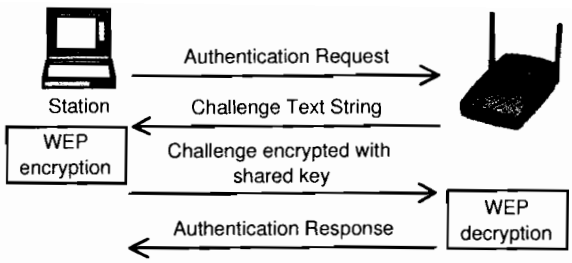

Figure 2.11: Shared Key authentication

The Shared key authentication system results in four messages communicated between the stations as opposed to an Open System that uses only two.

Deauthentication A station that wishes to leave a wireless network will first deauthenticate itself.

Privacy The privacy service ensures that information transmitted over the wireless medium is confidential. This is achieved by encrypting the contents of data messages.

Data Delivery Reliable data delivery is the responsibility of the data delivery service.

2.4.3 802.11 Link Layer Handoff

It was previously advanced that in order to effectively study Mobile IP handoffs, it is important to understand the properties of 802.11 link layer handoffs. This section will firstly describe what constitutes an 802.11 link layer handoff, and will then outline the mechanisms used in these handoffs.

The 802.11 standard [46] recognises that a station can undergo three types of movement in wireless LANs. These are listed below:

1. No Transition
   A station in this category is either stationary or is moving within a single BSS. For example, a moving station that remains within the coverage area of its associated AP experiences no transition.

2. BSS Transition
   In this case, a station moves from one BSS to another within the same ESS. A

station moving from one AP to another within the same ESS is an example of such movement. This class of movement is termed link layer handoff because it involves only link layer entities and mechanisms.

3. ESS Transition
   An ESS transition occurs when a station moves from a BSS in one ESS to a BSS in another ESS. The new ESS may be on a completely separate network. Therefore, this type of movement may result in the disturbance of upper layer connections. This type of movement motivates the need for Mobile IP.

A link layer handoff is a BSS transition where a station transfers its physical layer connectivity from one AP to another. Several services are invoked when a station performs a link layer handoff. The example in Figure 2.12 will be used to outline the sequence of steps that make up this process.

Handoff Initiation

As was shown in Figure 2.7, the quality of a wireless link gradually worsens as the communicating stations move further apart. If a station is associated with an AP and the link quality falls below a certain threshold, the station will initiate handoff to a "better" AP offering a higher quality link (if such an AP exists). A station usually uses physical layer parameters, such as signal strength or signal-to-noise ratio (SNR), as an indication of the wireless link quality. Periodic beacons from an AP allow a station to constantly evaluate these link parameters.

Figure 2.12 illustrates the link layer handoff initiation process as a station moves from AP 1 to AP 2. This figure was adapted from a technical bulletin [3] released by a specific wireless network card manufacturer. Specific details, such as threshold values, may differ depending on the particular 802.11 implementation. The following explanation provides a general overview of the processes that make up an 802.11 link layer handoff.

As the station in Figure 2.12 moves further from AP 1, the link SNR decreases. Conversely, as the station enters AP 2's coverage area, the signal strength of received beacons from AP 2 begins increasing. When the SNR of the station's link drops below a certain threshold (cell search threshold), the station begins searching for new candidate APs. The threshold is usually set so that a search begins before connectivity with the old AP is lost completely (a).

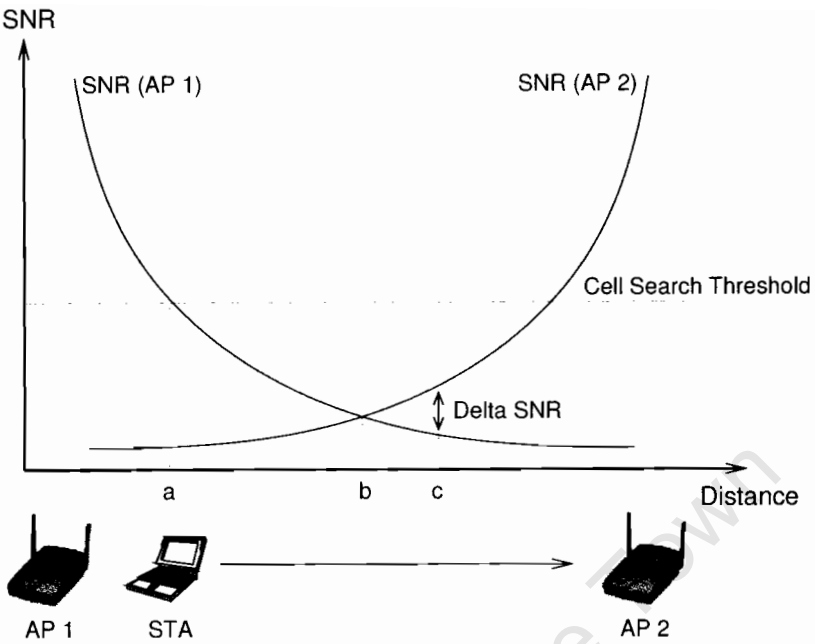

Figure 2.12: Movement between two overlapping APs

An alternative method of triggering handoff can be performed by waiting for a certain number of beacons to be missed or retransmission attempts to fail.

AP Discovery

A station searches for new candidate APs by performing a scan. Neighbouring access points usually operate on different frequencies to avoid interference[8]. The station must therefore perform a scan on all channels to discover any new APs. However, only APs on the same ESS as the station can be considered candidates. These APs are identified by their ESSID included in their broadcast beacons. The scan procedure is either performed actively or passively, as detailed below:

Passive scan In this mode, a station passively waits for beacon messages on a selected channel. After a time delay, the station moves to the next channel, eventually scanning all channels.

[8]Refer to Appendix C for more information on how channels should be assigned to neighbouring APs.

Active scan When a station performs an active scan, it broadcasts a *probe request* message on a selected channel and starts a timer. APs using this channel respond with a *probe response*. A probe response includes similar information to beacon messages, including ESSID and physical parameters. When the timer expires, the station repeats this process on the next channel. The probe responses are then prioritised according to signal strength to determine the best candidate AP.

As the station in Figure 2.12 moves towards AP 2, the SNR of probe responses from AP 2 increases. The station will select AP 2 when its SNR is higher than AP 1's SNR by a difference of delta SNR (at least). This is indicated in Figure 2.12 by point c. Once the most appropriate AP has been selected, the search phase of a link layer handoff ends. A station uses either an unsolicited beacon or a probe response to synchronise to the new AP. The station then begins connecting to the new AP.

Handoff Completion

The last stage of a link layer handoff is entered when the station attempts to reassociate with the new AP. This stage consists of two steps, authentication and reassociation. A connecting station must authenticate itself using the authentication service described in section 2.4.2 before reassociating with an access point.

According to the 802.11 standard, a station should disassociate with its old AP before attempting to associate with a new one. The standard also makes provision for preauthentication, whereby a station can authenticate itself with several APs in advance. A preauthenticated station does not need to authenticate itself during handoff, thus avoiding the four-way handshake overhead. This facility was introduced to speed up the reassociation process. However, both the disassociation and the preauthentication services are not observed in practical investigations into 802.11 handoff [76, 57].

Figure 2.13 outlines the three stages of an 802.11 link layer handoff. This diagram assumes that both active scanning and Open System authentication are used.

2.4.4 Analysis of Link Layer Handoff

Despite the thoroughness of the 802.11 standards, manufacturers have a great deal of latitude when designing and implementing 802.11 products. This section investigates the

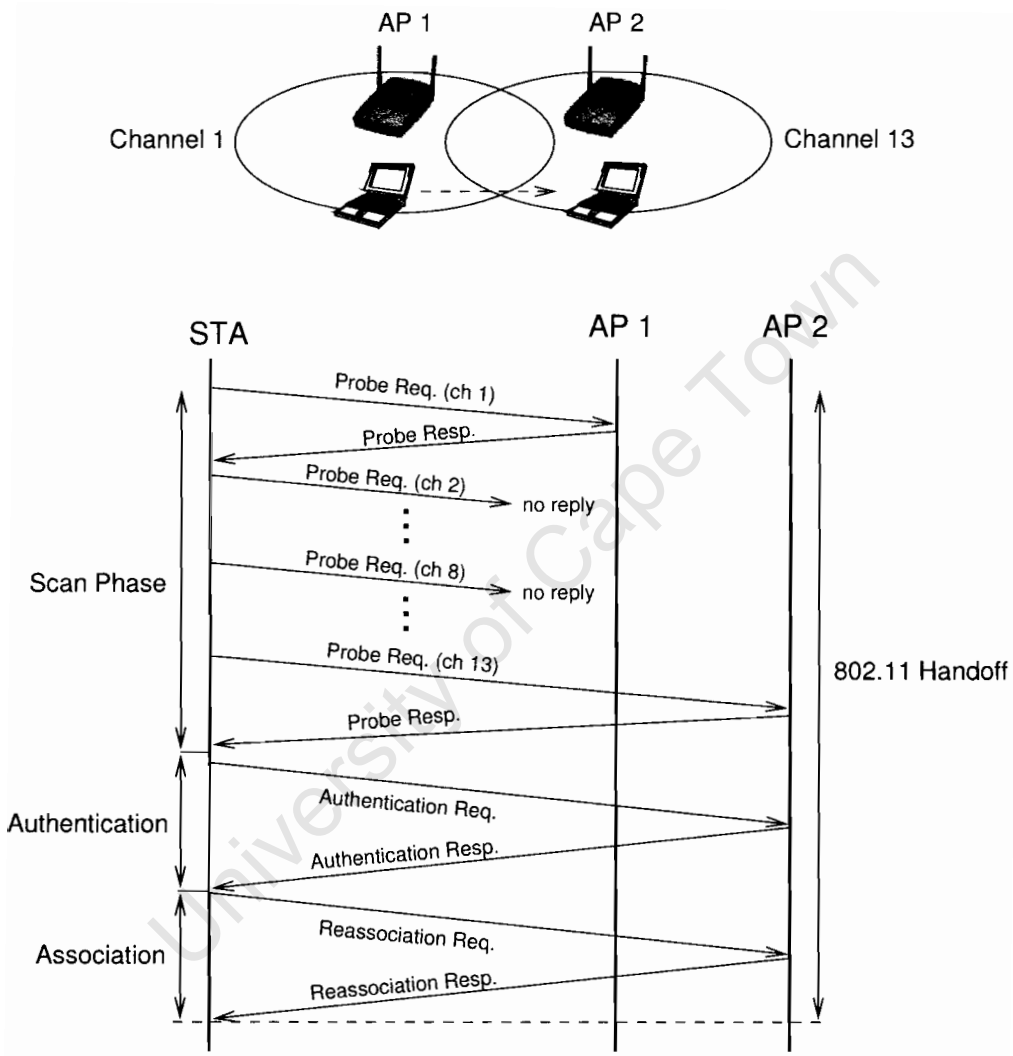

Figure 2.13: Link layer handoff timing diagram

properties and characteristics of practical 802.11 networks, with specific attention to 802.11 handoff. Due to the popularity and availability of 802.11b devices, most of these characteristics are inferred from tests with 802.11b equipment.

Two extensive studies have been conducted into 802.11b link layer handoffs. In 2002, 802.11b implementations from different manufacturers were investigated to determine the differences in their handoff performance [57]. Listed below are some of the most important characteristics that are discussed in these studies.

- 802.11 devices perform hard, "break before make" handoffs where a station disconnects from a previous AP before connecting to a new AP. Thus, during handoff, the station is temporarily disconnected and may experience a certain amount of traffic loss.

- The delay introduced by active scanning ("probing") procedures accounts for more than 90% of the total 802.11 handover time. Probe messages also account for 80% of the communication between a station and an AP during handoff. It is important to note that the selection of an AP is only definite once the scan process has been completed. This makes the prediction of a station's new AP very difficult and speculative at best until the end of the handoff process.

- Link layer handoff performance depends heavily on the specific implementation (e.g. device hardware, firmware and driver). Hardware from different manufacturers results in greatly varying handoff delays. Both 802.11 wireless cards and access points from different manufacturers were tested in these studies. Figures 2.14 and 2.15 illustrate a summary of the results obtained by this investigation. Figure 2.14 shows that the average handoff latency varies approximately between 50 ms and 400 ms depending on the card/AP configuration used.

- Another significant characteristic of link layer handoff is that performance fluctuates greatly even when considering a single card/AP configuration. This is evident from the relatively large standard deviations that were reported (Figure 2.15). These values range from 17 ms to 91 ms.

- It was observed that cards from different manufacturers follow slightly different handoff procedures. Sometimes these handoff message sequences do not conform with the behaviour specified by the 802.11 standard. For example, one of the tested cards

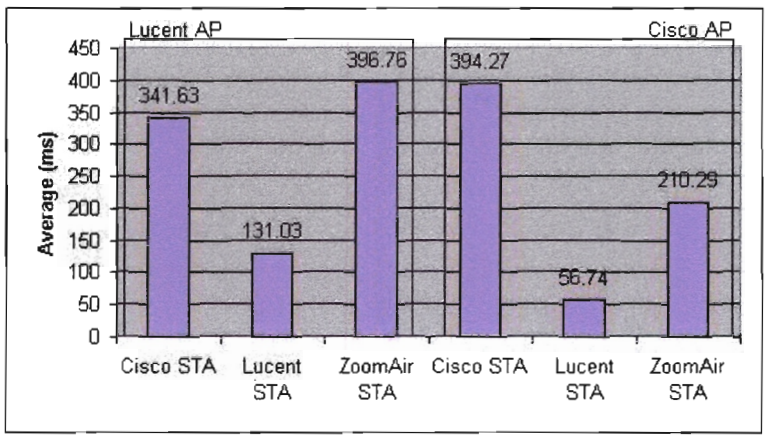

Figure 2.14: Average 802.11 handoff latency

sends a reassociation message prior to sending an authentication message. This is the reverse order of the sequence described in the 802.11 specification.

In 2003, a second study was carried out involving 802.11b handover and its effect on voice traffic [76]. The characteristics listed above have been confirmed by this and other studies [19, 59]. In this study, two hardware configurations were tested. 802.11 handoff is reported to last either 80 ms or 210 ms (approximately), depending on the card used. As mentioned above, the scanning phase was found to be the largest contributor to the total 802.11 handoff latency. In addition, the incorrect messaging sequence described above was also observed during 802.11 handoffs using both hardware configurations.

The results described above were derived by forcing a wireless card to perform several handoffs between different APs. However, even the way handoffs are triggered can have an effect on performance [10]. For example, one way of forcing a handoff is by disabling a station's current AP (by unplugging its power supply). The station is thus forced to begin searching for new candidates. An alternative method is to trigger a handoff by lowering the AP's transmit power level. Unfortunately, no quantitative comparisons have been conducted into the effects of these techniques.

In another study [59], both the effects that the number of stations and the raw bandwidth have on 802.11b handoffs are considered. It was discovered that a single idle station generally experiences a shorter handoff latency as compared to an active station. This is true for a station connected at any data rate (2, 5.5 and 11 Mbps) except for 1 Mbps.

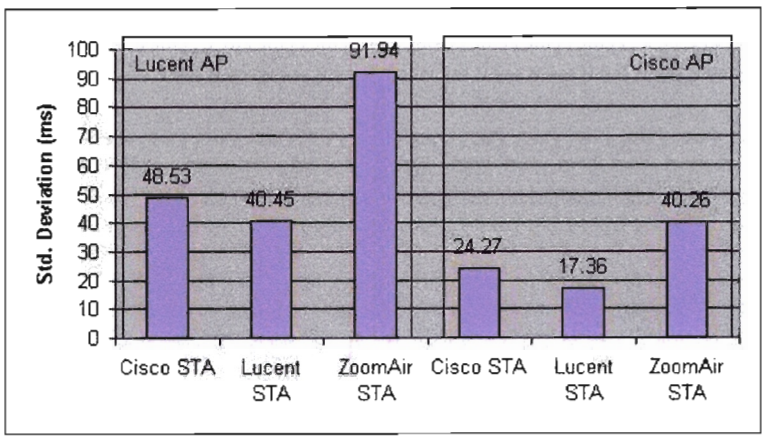

Figure 2.15: Standard deviation of 802.11 handoff latency

Handoff latency is also shorter for stations with low traffic levels as opposed to highly-active stations.

Another scenario is considered which includes 5 associated stations in addition to the station performing handoffs. The average handoff latency was observed to be over 30 times greater than in the single station scenario, lasting over 8 seconds. Similar results are reported in [76]. This demonstrates that link layer handoff latency is adversely affected by the number of associated stations in a wireless LAN. The main reason for this is that access to the shared wireless medium is delayed when the medium is shared among several competing stations [59, 76]. This effect is magnified when DCF RTS/CTS messages are used as they incur additional overhead.

The numerical results from the studies described above may differ due to the particular hardware configurations used. However, they all confirm the general characteristics of 802.11 (specifically 802.11b) handoffs introduced in this section.

2.4.5 Future Developments in 802 Standards

This section provides a brief overview of the areas and directions of development within the 802 standard bodies that are relevant to this study.

IEEE 802.11F – IAPP

The 802.11 standard does not specify the details of how stations and access points should perform 802.11 handoffs. This is apparent from the inconsistent results witnessed in current systems (discussed previously). Currently, most 802.11 access points do not implement any form of inter-access point coordination when a station roams between several APs in a wireless LAN. Some AP manufacturers however, have implemented their own proprietary protocols and mechanisms that attempt to optimise these link layer handoffs. Examples of such proprietary protocols are those used by Orinoco and Lucent systems [13, 5]. While these protocols may work sufficiently well between similar devices, they are incompatible with equipment from other manufacturers.

In June 2003, the IEEE 802.11 working group published the 802.11F document in an effort to rectify this situation [48]. This document outlines recommended practices for implementing an Inter-Access Point Protocol (IAPP). One of the main objectives of this publication is to ensure that 802.11 access points from different vendors are interoperable. The IAPP therefore establishes a means of inter-access point coordination, and attempts to streamline the interactions between these APs. Any information exchange between APs is achieved using the IP protocol because it is the most popular DS implementation.

Specifically, IAPP aims to optimise the 802.11 handoff process, thus allowing 802.11 stations to roam between different APs within an ESS more efficiently. This is achieved by transferring a station's context from one AP to another as it roams within a wireless LAN. A Remote Authentication Dial-in User Service (RADIUS) server allows the IP addresses of neighbouring APs within an ESS to be looked up based on their BSSID.

The IAPP supports the use of two types of messages by 802.11 APs. "Announce" messages inform other APs about a new active AP. Handover or "move" messages allow the responsibility for a station to be moved effectively from one AP to another. When a station reassociates with a new AP, these handover messages allow the station's context to be moved from the old AP to the new AP. An example of what such a context contains is security information that will speed up the reauthentication of the station at the new AP. The protocol also ensures that layer 2 devices such as switches and bridges are updated to accurately reflect the station's movement through a layer 2 update frame. IAPP handover is triggered by a station's Reassociation request [48].

Figure 2.16 [58] illustrates the message timing diagram of IAPP handoff messages when a station moves from an old AP to a new AP. IAPP is reactive in that it responds to

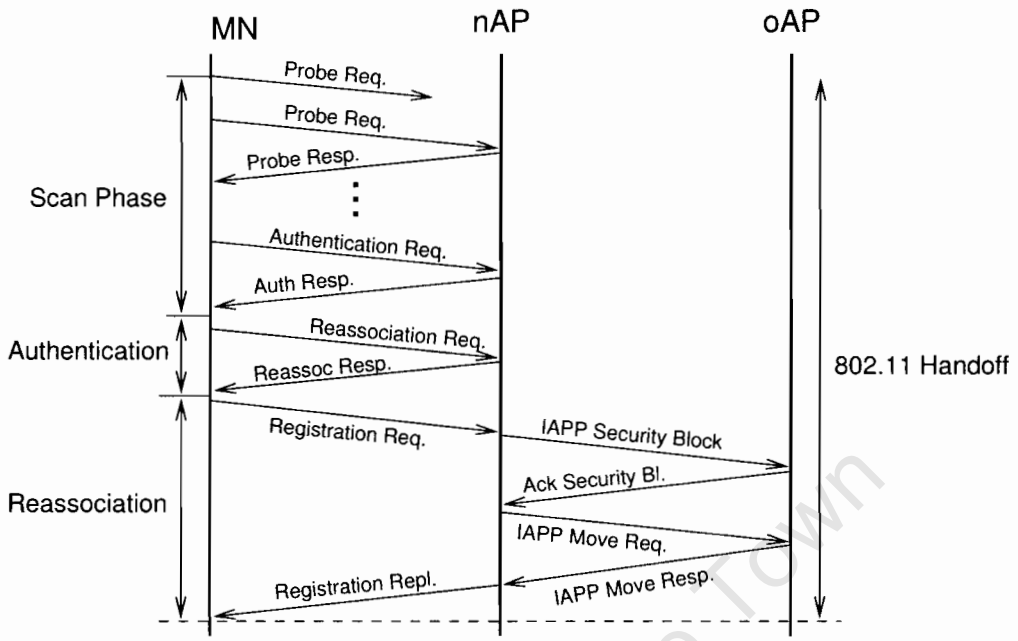

Figure 2.16: IAPP handover procedure

handoffs after they happen. As can be seen from the figure, IAPP has also added extra steps in the 802.11 handoff process. In order to counter-balance the delays associated with these steps, a station's context may be proactively cached with neighbouring APs that it might handoff to. A given AP can dynamically learn of the presence of neighbouring APs and verify their identities using the RADIUS server [48]. Proactively caching a station's context in this way dramatically reduces the 802.11 reassociation delay [58].

IAPP focuses on optimising link layer roaming. A station must still rely on DHCP or Mobile IP to receive a network layer address when moving between different IP networks. Further information regarding the IAPP protocol can be found in the IEEE standard and related documents [48, 58, 5].

IEEE 802.21 – Handover Interoperability

The IEEE 802.21 is a recently formed working group, established in February 2004 to deal with the issues of handover between different 802 technologies [9]. 802.21 is developing specifications to enable handover and interoperability between heterogeneous network types

including both wired and wireless 802 networks. An example of such handoffs would be between Ethernet (802.3) and 802.11. There has been a significant amount of discussion between the 802.21 and IETF DNA working groups[9]. This is in an effort to streamline and further integrate the development of both 802 link layer technologies and IP mechanisms.

The 802.21 working group is attempting to ensure that a host can perform an 802 handover without experiencing any interruption or delays. These handoffs will be suitably optimised such that handoffs between different technologies from different vendors will result in the same high level of performance. To achieve this, the 802.21 working group is currently developing a standard entitled "Media Independent Handover Services" [42].

The most important aspect of the 802.21 standard (relative to this study) is that it aims to "facilitate [the] optimisation of Mobile IP handover" [42]. In this respect, the standard will provide a greater interaction between 802 physical/link layers and the network layer. This will allow network layer protocols such as Mobile IP to use lower-layer information to improve their handover decisions [41]. Many of these issues will be discussed in greater depth in the next chapter.

IEEE 802.11e – QoS

Currently, 802.11 wireless LANs, like their wired Ethernet counterparts, do not support quality of service (QoS) guarantees. In order to meet the requirements of QoS-sensitive applications, the 802.11e standard is being developed within the 802.11 group. Example applications include the transport of voice, audio and video over 802.11 wireless networks, in addition to supporting the development of new multimedia applications.

To achieve these goals, 802.11e defines two new MAC sublayers. These are the Hybrid Coordination Function (HCF) and the Enhanced DCF (EDCF). The HCF and EDCF are extensions of PCF and DCF respectively. Both the HCF and EDCF modes are able to differentiate traffic flows, allowing some traffic to be given priority over others. This is in contrast to the existing DCF and PCF modes that treat all traffic equally. Both the EDCF and HCF modes define eight priority levels. The details of how these MAC scheduling schemes achieve these service levels is beyond the scope of this document. Further information is presented in [40, 78].

[9]Further information is available at: `http://www.ieee802.org/21/` and `http://www.drizzle.com/~aboba/IEEE/`

802.11 Handoff Optimisations

Several means of improving 802.11 handoff performance have been proposed in the literature. While, an in depth discussion of these techniques is beyond the scope of this study, a brief overview is given below.

Most of the 802.11 handoff latency is contributed by the scan phase. One way to minimise this delay is by optimising the active probing mechanisms. Instead of scanning all channels sequentially, a station can initially scan the channels that neighbouring APs are most likely using. Most AP vendors specify the channel separation that adjacent APs should use to prevent interference [74]. A station could use this knowledge to establish the order in which to scan available channels.

Alternatively, a list of channels being used by neighbouring APs (known as a "candidate list") could be provided to the station by the station's current AP. This implies that APs are able to acquire information about their neighbouring APs through some means such as IAPP [10]. A station can also avoid performing a full active scan during handoff by performing the scan (either partially or completely) while the station is idle. These techniques, along with further enhancements to the scanning phase, are discussed in other studies [71].

802.11 handoff performance would be improved if wireless card hardware/firmware behaviour during handoff complied with the IEEE 802.11 standard. Incorrect message sequences result in retransmissions and waste both time and resources during a handoff [76].

Chapter 3

Mobile IP Handoff System Overview

3.1 Introduction

The wireless network architecture under investigation has been previously defined and explained in detail. This chapter will proceed to focus on how the Mobile IP handoff process can be improved. The chapter begins with a brief outline of Mobile IP handoff, along with a description of how 802.11 influences its performance. The foundation of this study is that Mobile IP handoff latency can be reduced significantly through the use of movement detection optimisations. Special attention will therefore be given to the movement detection procedure, focusing specifically on the factors that make it inefficient. Mobile IPv4 movement detection will be discussed along with some of the additional mechanisms that have been incorporated into Mobile IPv6. Later chapters dealing with the design of various optimisations will build onto the information contained in these sections. The reason for discussing both protocol versions is to allow a theoretical performance comparison to be drawn between the two. The choice of Mobile IP version used in subsequent chapters will be based on this comparison. Thereafter, the principles of micro/macromobility will be introduced. These protocols ensure that registration delays are minimised, effectively isolating the movement detection process. Finally, high-level designs for several movement detection optimisations described in the literature will be developed. These techniques will pave the way for the introduction of the author's proposed hybrid movement detection technique, examined at the end of this chapter.

Before Mobile IP handoff is described, it is important to note that handoffs in general (layer 2 or 3) may be classified into several different groups. Different types of handoff

have different implications. These must be understood before a discussion involving specific technologies can take place.

A mobile node (MN) usually connects to an IP network via an "access entity" such as an 802.11 AP (layer 2) or a Mobile IPv4 foreign agent (layer 3). Handoff latency is defined as the time delay between the last packet/frame received through a previous access entity and the first packet/frame received through a new access entity. A *fast* or *low latency* handoff specifically minimises the time interval during which a MN is unable to send or receive information. A *smooth* handoff minimises the data loss that occurs during a handoff. A smooth handoff does not automatically imply a fast handoff. For example, a smooth handoff can be performed despite significant latencies by using buffers to minimise loss of data [29]. A *seamless* handoff is both fast and smooth and this represents the ultimate goal for any mobility management system.

Handoffs can also be divided into two categories: *hard* and *soft*. A hard handoff (sometimes referred to as "break-before-make") occurs when a MN is only able to communicate through one access entity at any stage in the handoff. On the other hand, a MN performing a soft handoff ("make-before-break") will be able to access the network through more than one access entity simultaneously, given that a region exists where coverage areas overlap (Figure 3.1 (a)). For example, during a soft handoff a MN can configure a new CoA through the new network while still receiving packets through the old entity. It will be shown that soft handoffs cannot be performed when using technologies such as 802.11. Furthermore, soft handoffs are not possible when coverage areas do not overlap (b).

When the coverage areas of wireless access technologies do not overlap, regions exist where a terminal loses connectivity. The length of time that connectivity is lost during a handoff is affected by factors such as the terminal's movement pattern, speed and distance of the uncovered segment. In order to avoid such complications, this study will assume that all wireless LANs under consideration overlap to some extent, unless otherwise stated[1].

Figure 3.1 illustrates that a Mobile IP handoff decision can be influenced by many diverse factors. It is possible that a MN may find itself in the overlap region (a) with more than one available access technology. For example, the MN may have access to a GPRS network through one interface and a wireless LAN through another. Ideally, factors such as available bandwidth and cost associated with each network should be taken into account

[1]The wireless LANs depicted in various figures are often NOT shown as overlapping for the sake of clarity.

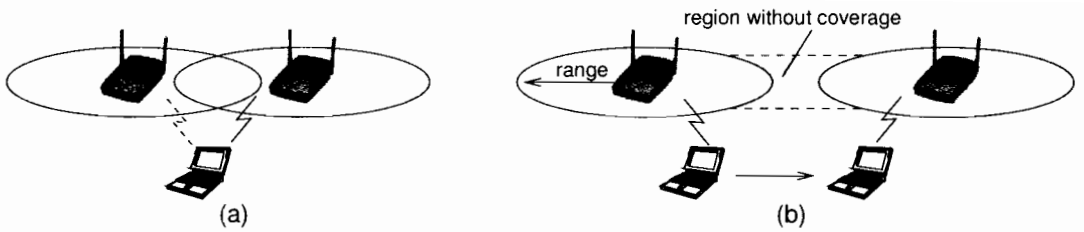

Figure 3.1: Overlapping (a) and Non-overlapping access technologies (b)

before performing a handoff. However, for the purposes of this study, Mobile IP handoff decisions will aim only to minimise disruptions to a MN's IP connectivity.

3.2 Mobile IP Handoff Overview

A Mobile IP handoff occurs when a mobile node acquires a new CoA as a result of changing its current access router or mobility agent. This typically occurs when a MN moves from one IP network to another. Figure 3.2 will be used to explore the Mobile IP handoff process further. It depicts a MN performing a Mobile IP handoff between two foreign 802.11 wireless LANs (movement 2).

The Mobile IPv4 and Mobile IPv6 handoff procedures are generally very similar. For simplicity, the following description refers only to Mobile IPv4 entities. The differences in handoff mechanisms between the two versions of Mobile IP will be discussed in later sections.

Mobile IP handoff can be divided into the following three main processes:

- Link layer handoff
- Movement detection
- Registration

Chapter 2 described how an 802.11 station roams within a wireless LAN[2], using signal quality metrics to initiate layer 2 handoffs between different APs. This is indicated in

[2]For simplicity, it is initially assumed that a given wireless LAN will consist of only one ESS. The two terms are therefore used synonymously.

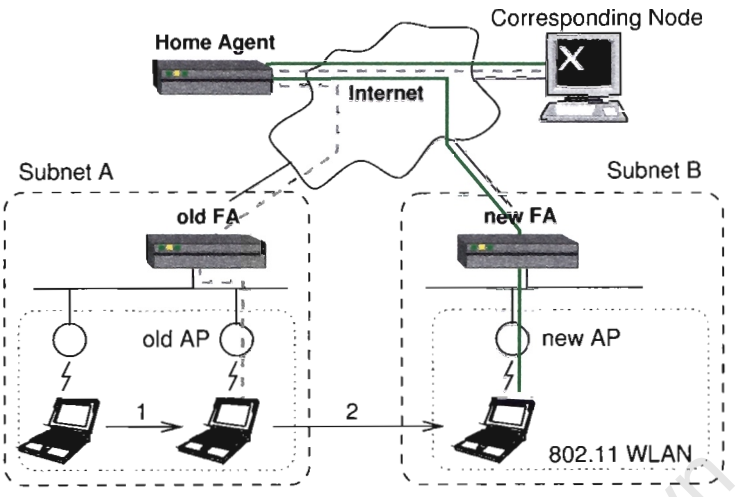

Figure 3.2: Mobile IP handoff between two wireless LANs

Figure 3.2 by movement 1. However, link layer handoff can be initiated in another way. During movement 2, the MN's 802.11 device will not be able to automatically handoff between the old and new APs because they are on different ESSs. In order for a MN to associate itself with an AP on a different wireless LAN (using a different ESSID), some external input is needed from a user or operating system to specify the new ESSID. When this input is provided, a link layer handoff is initiated and performed as described in Chapter 2 using the new ESSID.

Strictly speaking, link layer handoff is not part of Mobile IP handoff because it involves only layer 2 mechanisms. However, within the architecture under consideration, link layer handoff always precedes Mobile IP handoff and therefore contributes to the total handoff delay experienced by applications. Furthermore, 802.11 is only capable of supporting hard handoffs because a station may only be associated with one AP at a time. This factor implies that Mobile IP is also restricted to performing hard handoffs when operating over 802.11. The reason for this is that an 802.11 link defines the IP network that a MN is connected to, and by extension, allows only on-link mobility agents to be detected. Therefore, because Mobile IP handoffs between 802.11 wireless LANs are hard, the processes listed above must be performed in sequence and cannot be run in parallel or overlapped. It is clear that because the three stages of a Mobile IP handoff are performed sequentially, the total handoff latency is the sum of the delays of each individual stage.

Figure 3.3 illustrates the complete messaging sequence of a Mobile IP handoff that is performed for movement 2 in Figure 3.2 above. The diagram depicts the interactions that take place when a MN connects to a new foreign IP subnet for each of the above stages. These interactions involve the MN, the new access point (nAP), the new foreign agent (nFA) and the MN's home agent (HA).

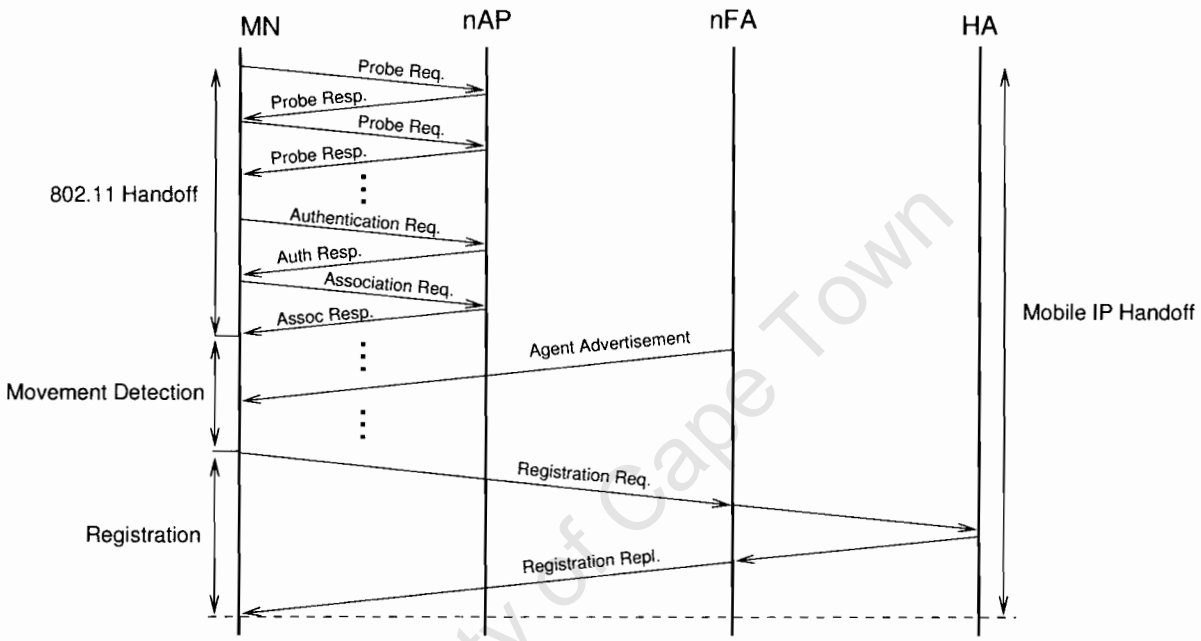

Figure 3.3: Mobile IP and 802.11 handoff sequence

Link layer handoff is completed once the MN's 802.11 device has been associated with the nAP and a link layer connection to the new network is established. A MN subsequently uses movement detection mechanisms to discover the presence of any new agents and recognise whether or not IP-level movement has taken place.

As was mentioned in the previous chapter, movement detection is performed through the reception of periodic advertisements. These advertisements are IP packets broadcast by mobility agents which contain information relevant to a visiting MN. Movement detection can thus be further subdivided into two stages: *discovery* and *selection*. The discovery phase is the interval between link layer handoff completion and the reception of the first new advertisement on the new link[3]. The selection stage follows immediately afterwards,

[3]This phase is termed "Agent Discovery" in Mobile IPv4

where the Mobile IP layer decides whether or not to handoff to the new agent. This decision is usually made according to a particular policy, using information contained in the received advertisements.

Once movement detection has been concluded and a new agent has been selected, the MN must configure its new CoA. Although this is a critical step in a handoff, it has not been included in the list above because it can be performed without significant delay. For example, a Mobile IPv4 host can use information gained during movement detection to immediately configure its own CoA. However, if external (stateless) mechanisms are used such as DHCP, the acquisition of a CoA would introduce significant delays between movement detection and registration. In addition, the performance of IPv6 Duplicate Address Detection (DAD) *may* also hamper the configuration of a CoA. It will be shown that in these cases, CoA configuration forms an additional step performed after movement detection.

In the last stage of Mobile IP handoff, registration is performed using the newly configured CoA. The MN transmits a registration request through the nFA back to its home agent[4]. An IPv6 MN transmits additional binding updates to its corresponding nodes. When the home agent successfully updates the MN's binding, a registration response is sent back to the MN (via the foreign agent if applicable). At this stage, the MN's traffic is no longer sent to its previous CoA, shown in Figure 3.2 as a dotted grey line. Traffic is instead forwarded to the MN's new CoA, indicated by the solid green line.

Whenever a MN moves from one network to another, authentication must be performed to some extent. The MN may have to be authenticated on several levels. A MN uses registration messages to carry authentication information back to the home agent which ensures that bindings are updated securely. Thus, authentication delays are included within the registration delay (just as the authentication service is incorporated into the 802.11 handoff latency). However, when an Authorisation, Authentication and Accounting (AAA) protocol is used (e.g. RADIUS), additional signalling is needed to authenticate the MN. This authentication stage adds further delays to the Mobile IP handoff process. The issues associated with MN authentication will not be investigated in this study.

[4]When using a co-located CoA, a MN *may* register directly with home agent (bypassing the FA).

3.3 Movement Detection

Due to the layer independence between Mobile IP and the link layer, a MN relies purely on IP-based mechanisms to detect when it has moved away from its current mobility agent/access router. Movement detection also allows new agents/routers to be discovered on a new link and provides the MN with the corresponding IP configuration information (such as network prefix, CoA etc.). The mechanisms that support Mobile IPv4 and Mobile IPv6 movement detection are investigated below.

3.3.1 Mobile IPv4

Mobile IPv4 uses Agent Discovery mechanisms to perform movement detection. Agent Discovery messages extend standard Internet Control Message Protocol (ICMP) Router Discovery messages to include mobility-specific information. Agent Discovery, like ICMP Router Discovery [32], uses agent advertisement and solicitation messages to detect new mobility agents. The packet format of a mobility advertisement extension is shown in Figure 3.4.

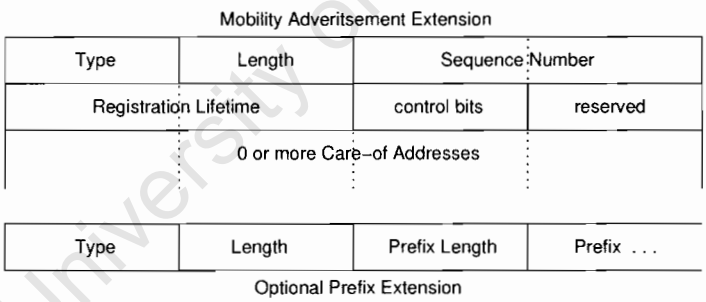

Figure 3.4: Mobility agent advertisement extension (with optional prefix extension)

Mobile IPv4 movement detection relies on the periodic broadcasting of agent advertisements by all mobility agents[5]. In addition to the standard ICMP fields such as on-link router address and lifetime, an agent advertisement also includes a registration lifetime. The registration lifetime is the maximum time that the MN is allowed to remain registered

[5]These broadcasts are addressed to the "all systems on this link" multicast address 224.0.0.1 or broadcast address 255.255.255.255 [28]

to a particular mobility agent. A registration with a particular mobility agent is essentially lease-based, and if necessary a MN can re-register with its current agent when the registration lifetime expires. It is important to note that the registration lifetime is independent from the ICMP lifetime, which defines the advertisement's period of validity. The sequence number field contains a counter that is incremented after each broadcast and prevents advertisement duplication. A foreign agent may also include a list of foreign agent CoAs available to a visiting MN within the CoA field. In addition to the mobility advertisement extension, a mobility agent may append a prefix-length extension to its advertisements. This extension indicates the network prefix part of any supplied IP addresses and can be used to identify the local network.

In the first Mobile IPv4 draft specification (RFC 2002 [27]), the recommended maximum rate for broadcasting advertisements is once per second. This RFC was rendered obsolete by RFC 3220 [28] which, in turn, was replaced by RFC 3344 [28]. Both updated RFCs have removed this maximum advertisement rate limitation. Instead, a more general guideline is offered. This recommends that the advertisement broadcast rate should be limited in such a way that it does not consume significant network bandwidth.

Irrespective of advertisement rate, a MN that does not wish to wait for a periodic advertisement may request a unicast agent advertisement by broadcasting an agent solicitation. All mobility agents that receive a solicitation will reply with an advertisement. However, the ICMP Router Discovery specification [32] imposes certain delays on an agent solicitation/advertisement exchange. A MN should wait a random time up to one second before sending a solicitation and a mobility agent must delay solicited advertisements up to two seconds. These delays prevent the synchronisation of solicitations from several MNs, which may occur if several MNs are switched on at the same time. They also prevent synchronisation of replied advertisements from several agents and allow several closely-spaced solicitations to be answered with one advertisement [32].

A MN must limit the rate at which it broadcasts agent solicitations. When searching for an agent, a MN may broadcast solicitations at a maximum rate of one per second. After three solicitations have been sent, this rate should be reduced. Furthermore, a mobility agent must respond to agent solicitations even when the source address is not valid on the local network (e.g. a visiting MN). For more information, refer to the Mobile IPv4 specification [28].

There are two standard movement detection algorithms offered by Mobile IPv4 [28]. The

first algorithm is based on the ICMP lifetime field included in the standard ICMP portion of an agent advertisement. This advertisement lifetime is the maximum time that an advertisement may be considered valid in the absence of subsequent advertisements. Therefore, if a MN does not receive a subsequent advertisement from the same agent within the specified lifetime, the MN should consider the current mobility agent unreachable. This *may* imply that the MN has moved to a new network. If, at this stage, the MN has not cached any recent advertisements from other agents, it should begin actively searching for new agents by broadcasting an agent solicitation. The interval between successive agent advertisements is at most one third of the advertisement lifetime. In other words, a MN will miss three advertisements before the current foreign agent is assumed unreachable. For example, when using the maximum rate recommended by RFC 2002, advertisements are broadcast once per second. In this case, the MN only assumes that movement has occurred after three seconds have passed without an advertisement ($3 \times$ Advertisement Interval). This significant delay prompted the maximum advertisement rate recommendation to be lifted in later Mobile IPv4 specifications so that higher rates can be used.

Figure 3.5 illustrates how this algorithm behaves in an 802.11 wireless LAN environment, such as the scenario shown in Figure 3.2. While connected to the first subnet (subnet A), the MN periodically receives advertisements from foreign agent A ("old" FA) . Once the MN performs a link layer handoff to subnet B, advertisements are no longer received from the old FA. Instead, the MN discovers the new FA when it receives an advertisement on the new link. The interval between wireless link establishment and reception of the first new agent advertisement is the discovery phase of movement detection, and is indicated in Figure 3.5 below. At this stage, the new FA is not selected yet, because the old FA has not been confirmed as "unreachable". Once three old FA advertisements have been missed, the new FA is selected, the selection phase is completed and movement detection ends.

The selection policy described above aims to delay Mobile IP handoff until it is absolutely necessary. A MN's registration on a particular network should be preserved for as long as possible in order to avoid the large signalling overheads and delays that a Mobile IP handoff introduces. This selection policy has been termed Lazy Cell Switching (LCS) or lazy-binding [35, 11].

The Mobile IP specifications have defined an alternative movement detection selection policy called Eager Cell Switching (ECS) or eager-binding [35]. When using this policy, a MN will consider any previously unheard router advertisement to be an indication that a network-layer handoff has taken place. Instead of waiting for a timeout to occur due to

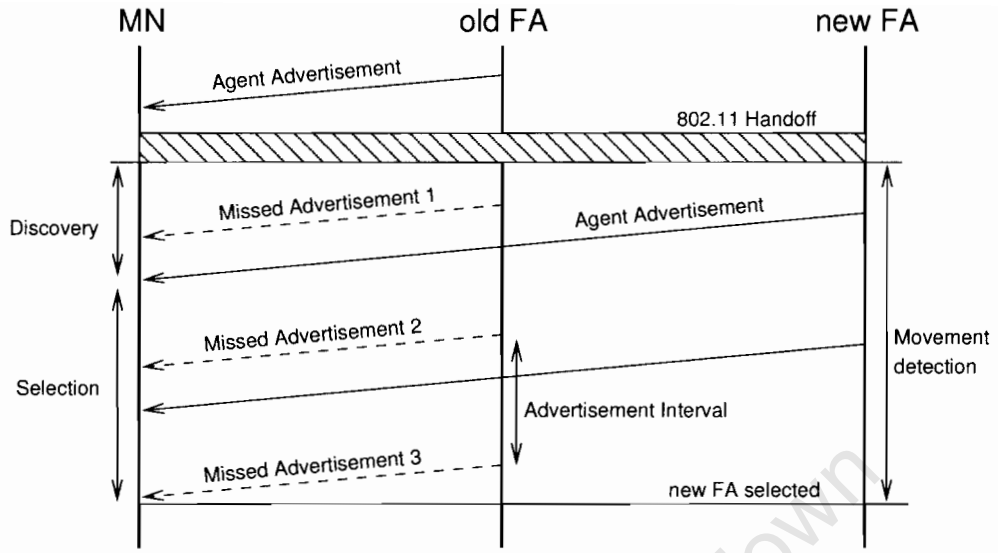

Figure 3.5: Default movement detection algorithm (lazy-binding)

missed advertisements, movement is detected after the first new advertisement. An eager-binding strategy therefore removes the entire selection phase delay. This policy assumes frequent location changes and straight line terminal movement. Under these conditions, eager-binding will significantly improve handoff performance over the lazy-binding policy because selection delays are avoided. This improvement can be seen in Figure 3.6.

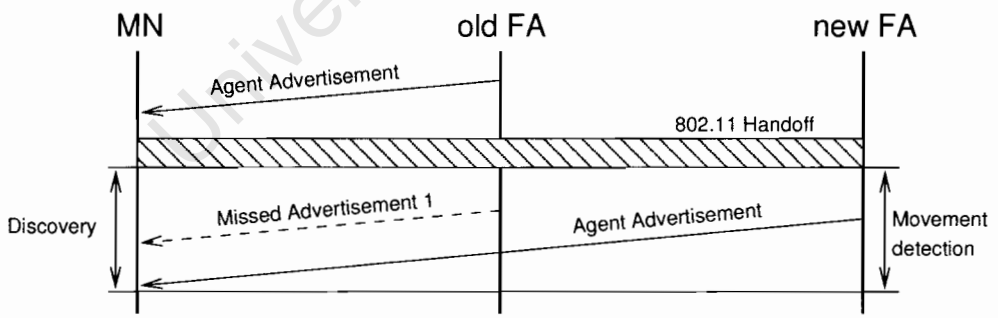

Figure 3.6: Movement detection with an eager-binding selection policy

Unfortunately, the eager-binding policy has certain drawbacks. As was mentioned in the previous chapter, mobility agents are susceptible to becoming bottlenecks because they

handle all their mobile nodes' traffic. One way to prevent this is by deploying several agents on a particular IP subnet that will share the load. However, using an eager-binding policy within a network serviced by several mobility agents may lead to unnecessary handoffs. This issue will be explained further in section 3.6.7.

The final Mobile IPv4 movement detection algorithm makes use of the optional network prefix-length extension that may be included in agent advertisements. If an advertisement is received from a new foreign agent advertising a new network prefix, then the MN should assume it has moved to a new network. This algorithm is somewhat less reliable than the previous techniques because mobility agents are able to omit prefix information from their unsolicited advertisements. In such a case, MNs cannot use this algorithm. Due to these factors, this algorithm will not be studied in further detail.

It is important to note that the Mobile IPv4 draft specifications allow existing movement detection mechanisms to be supplemented with other (link or network layer) information or mechanisms. Obviously, the details of this information is dependent on the characteristics of the underlying link layer. These techniques can be used to speed up the movement detection process.

3.3.2 Mobile IPv6

In Mobile IPv6, movement detection is based primarily on the IPv6 Neighbour Discovery protocol [61]. Neighbour Discovery includes mechanisms such as Router Discovery and Neighbour Unreachability Detection (NUD). Neighbour Discovery essentially incorporates the functions provided by the IPv4 ARP and ICMP Router Discovery protocols. Neighbour solicitation and advertisement (NA/NS) messages are used to detect a neighbouring node's presence on the link, along with its link layer address. Neighbour Discovery also defines router advertisement and solicitation (RA/RS) messages that allow on-link routers to be discovered and provide nodes with appropriate IP information such as network prefixes. This information can be used by a node to configure its IP address[6]. A standard IPv6 router will broadcast router advertisements periodically so that hosts on the link will be able to learn of its presence within a few minutes [61]. Much like Mobile IPv4, these periodic advertisements form the basis of Mobile IPv6 movement detection.

[6]Router advertisements are similar to Mobile IPv4 agent advertisements. Refer to the Mobile IPv6 specification [51] for details.

Mobile IPv6 extends Neighbour Discovery in certain areas to improve support for mobility [51]. For example, a standard router typically includes its link-local address in the source address field of its router advertisements. Mobile IPv6 allows an access router to advertise its global IP address in router advertisements within an optional Prefix Information extension. This global address is useful to MNs because it is unique and can be used to detect new access routers reliably.

Router advertisements may also include an Advertisement Interval option which informs a MN of how often it can expect to receive unsolicited advertisements from the access router. The Neighbour Discovery specification [61] limits the interval between successive advertisements to a minimum (randomly distributed) between 3 to 4 seconds. This relatively low rate may be adequate for large fixed networks, however MNs require configuration information within much shorter time intervals. In order to allow faster movement detection, this rate may be significantly increased for routers that support Mobile IPv6 nodes. The minimum allowed interval used by access routers is between 0.03 and 0.07 seconds (50 ms on average) [51]. The Mobile IPv6 specification warns that these higher rates should be used judiciously and the characteristics of the link should be taken into account. For example, in certain wireless links, such as 802.11, high broadcast rates waste considerable network resources.

A router solicitation and replied router advertisement (RS/RA) exchange between a MN and an access router includes a number of built-in delays (for the same reasons that they are included in ICMP Router Discovery). A MN should wait for a random interval (0–1 second) before transmitting a router solicitation. Likewise, a router must include a random delay (0–500 ms) before replying with a RA [20].

Although a number of extensions have been incorporated into the default Mobile IPv6 movement detection process, they are based on similar principles as Mobile IPv4. Furthermore, Mobile IPv6 does not explicitly perform faster movement detection than Mobile IPv4. For these reasons, only the mechanisms that support Mobile IPv6 movement detection are described above. For more information on the IPv6 movement detection process itself, refer to Appendix A.

3.4 Comparison of MIPv6 and MIPv4 Handoff

Mobile IPv6 has used the experience gained in the development of Mobile IPv4 to improve its efficiency and integration. As a result, a number of enhancements have been incorporated into Mobile IPv6. Although they are both functionally similar, Mobile IPv6 mechanisms make more extensive use of standard IPv6 functions and entities. Mobile IPv4 on the other hand, was developed as an addition to IPv4 and is less integrated. All these factors ensure that Mobile IPv6 is more efficient than Mobile IPv4. However, how these enhancements translate into quantitative time savings during handoff remains unclear.

In Mobile IPv6 networks, deployment of dedicated foreign agents is unnecessary and their functionality is performed by standard IPv6 access routers. In Mobile IPv4, a MN either uses a foreign agent's address (or one of its addresses) or alternatively relies on an external mechanism such as DHCP to configure a new CoA. An IPv6 MN can also use external stateless mechanisms, but has the option of configuring its own CoA using Address Autoconfiguration. When a MN uses an external stateless mechanism to configure its (co-located) CoA, it is subjected to the delays imposed by that mechanism. For example, a MN using DHCP has to discover and query a local DHCP server. The associated messaging overhead may introduce additional delays before a CoA is received. In contrast, an IPv4 MN can immediately configure a foreign agent CoA from information included in agent advertisements (using a foreign agent CoA).

Because foreign agents do not exist in Mobile IPv6, an IPv6 MN cannot receive a "foreign agent CoA". Instead, IPv6 Address Autoconfiguration allows a MN to efficiently configure its own CoA. The main drawback of using Address Autoconfiguration is that Duplicate Address Detection (DAD) must be performed for all global IP addresses. DAD is an extremely inefficient process because the MN must transmit several broadcasts and wait for possible replies before ascertaining that a particular address is not in use. Furthermore, a valid link-local address is needed in order to configure a global IPv6 address. This means that DAD would have to be performed twice, for the link-local and global addresses respectively. In order to overcome this drawback, the Address Autoconfiguration draft [62] specifies that DAD for a link-local address can be performed in parallel to global IP address configuration. Despite this improvement, DAD still remains a time-consuming process. The need for an optimised DAD procedure that avoids the disruption caused by conventional DAD has been recognised within the IETF DNA[7] working group. The framework requirements

[7]Detecting Network Attachment

for an optimised DAD mechanism have been specified with the aim of supporting real-time services [65, 64]. These two optimisations illustrate that DAD delays can be reduced, although they are not completely eliminated. At the time of writing, optimised DAD has not been fully developed or tested.

Mobile IPv6 uses slightly different techniques from Mobile IPv4 to perform movement detection. However, they both rely on the same mechanisms, such as periodic advertisements. Both protocols allow high advertisement rates in order to promote faster movement detection. They also both allow movement detection to be influenced by link layer information, although neither specification describes a framework for achieving this. Mobile IPv6 includes two additional steps in the movement detection process. These are the confirmation of AR reachability (NUD) and DAD, both of which can introduce significant time delays. An IPv6 MN must also use two separate NS/NA and RS/RA exchanges before completing movement detection (refer to Appendix A). On the other hand, Mobile IPv4 includes larger random delays in solicitation/advertisement exchanges.

In conclusion, Mobile IPv6 contains a number of facilities that streamline mobility management as compared to Mobile IPv4. However in the area of handoff, and specifically movement detection, both of these protocols use similar techniques. As a result, there are no theoretical factors that indicate one protocol's clear superiority. In addition, no quantitative results could be found in the literature that compare Mobile IPv4 and Mobile IPv6 handoff performance. It is therefore extremely difficult to isolate with certainty which protocol results in faster and more efficient handoff performance. These issues provide an avenue for further study.

Having ascertained this, Mobile IPv4 is a more attractive system for the purposes of this study because its movement detection process is both simpler and more well-defined. For instance, the time-consuming DAD and NUD mechanisms are not used in Mobile IPv4. Automatic CoA configuration is also less complicated when a MN uses an IPv4 foreign agent CoA.

3.5 Micromobility

One of the inherent flaws in Mobile IP handoff is the registration procedure's lack of scalability. When highly mobile terminals move between networks with small coverage areas, such as 802.11 wireless LANs, handoffs are performed frequently. During each of

these handoffs, registration messages will most likely travel through the Internet and may be subjected to significant delays or lost completely as a result. In addition, Mobile IP will perform the same complete registration irrespective of whether the MN experiences large or small-scale movement. These issues worsen when considering large numbers of MNs due to the increased registration signalling load.

The "principle of locality" states that a terminal will usually experience local movement between neighbouring networks [15]. This principle allows IP mobility to be divided into *macromobility* and *micromobility*. This separation is achieved by using a hierarchical access network infrastructure, called an administrative *domain*, that consists of several IP subnets. A micromobility protocol therefore deals with local, inter-subnet movement within the domain, while a macromobility protocol is responsible for inter-domain mobility. A domain gateway serves as the interface between the two, providing access to the Internet.

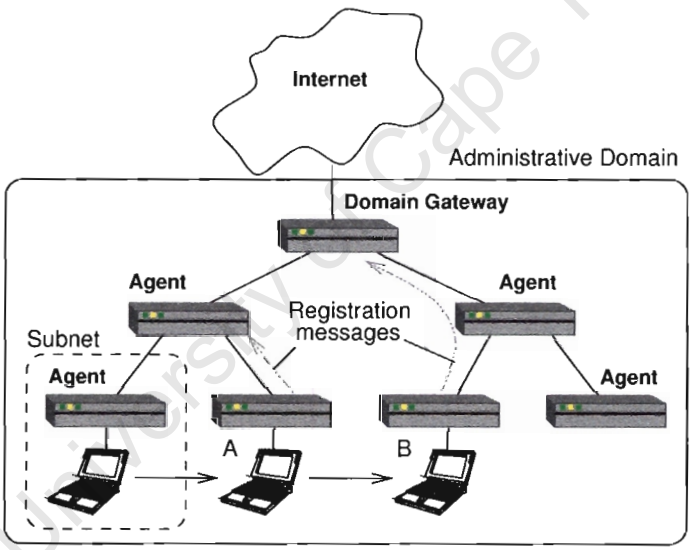

Figure 3.7: Micro/macromobility architecture

Within the domain, a hierarchical tree of mobility agents can be established, as depicted in Figure 3.7. When a MN node moves from one IP subnet to another within the domain, binding updates are confined to the domain instead of travelling through the Internet. For such intra-domain mobility, "regional registration" can be performed, where binding updates need only be relayed to local mobility agents. This significantly reduces the registration delay component of a Mobile IP handoff. For example, when the MN in Figure 3.7

performs a handoff between subnet A and B, registration messages only travel as far as the domain gateway.

Many different micromobility protocols have been developed such as Hierarchical Mobile IP, Cellular IP and HAWAII [68]. Most protocols follow the same structure illustrated in Figure 3.7 and have similar general characteristics. In most cases, these micromobility protocols were designed to interoperate with Mobile IP and rely on Mobile IP to provide macromobility management. A detailed investigation of these different protocols is beyond the scope of this study. For more information, Reinbold and Bonaventure [68] provide an in-depth comparison of different micromobility protocols.

The wireless access network model presented in previous chapters can now be extended to support micromobility as shown in Figure 3.8.

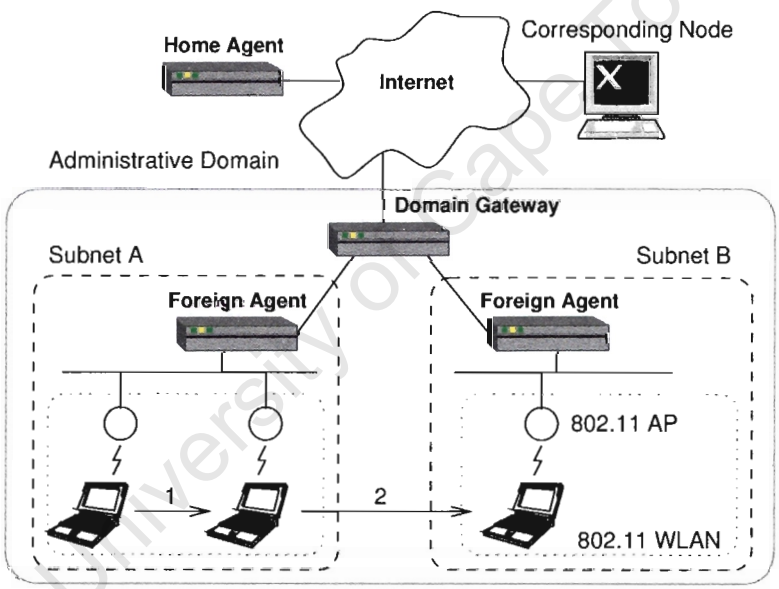

Figure 3.8: Hierarchical wireless access network model

3.6 Movement Detection Optimisations

It has been shown that Mobile IP handoff can be subdivided into various stages and that optimisations to these stages result in lower handoff delays. Thus far, only enhancements

to the link layer handoff and registration processes have been described. Recall that the characteristics of layer 2 handoffs are defined by the particular link layer technology. Specifically, 802.11 mechanisms are implemented in the hardware and firmware of a particular device and have to conform to the IEEE 802.11 standard. They are therefore extremely difficult to modify or optimise when researching current commercial devices. However, several development groups such as the 802 "Handover Interoperability" group (802.21) and 802.11F (IAPP) are currently investigating techniques to improve 802.11 handoff performance. In addition, the previous section illustrated that Mobile IP registration delays can be minimised within an administrative domain by using a hierarchical micromobility protocol. Despite all these enhancements, the movement detection process (in both MIPv4 and MIPv6) remains highly inefficient and contributes significantly to Mobile IP handoff latency. It is for this reason that the remainder of this study focuses on optimising movement detection.

3.6.1 Advertisement-Based Movement Detection Characteristics

The delays introduced by the movement detection process essentially arise from the layer independence between Mobile IP and the link layer. Generic movement detection is achieved through the reception of periodic IP advertisements. Movement detection performance is therefore directly linked to the advertisement rate. Higher advertisement broadcast rates (i.e. shorter advertisement lifetime intervals) result in faster movement detection.

However, in the generic advertisement-based movement detection schemes mentioned above, there is an inherent trade-off between the bandwidth used by advertisements and the movement detection performance. The higher the rate that periodic advertisements are broadcast, the more bandwidth is "wasted" by these messages. This inverse relationship is especially pertinent when considering wireless networks such as 802.11b where link bandwidth may be reduced when low data rates are employed, as illustrated in Chapter 2. These factors are even more pronounced within a micromobility architecture involving several agents that all broadcast their advertisements at high rates.

Discovery of a new router/agent through its advertisements is also related to the advertisement rate. MN movement is independent from an access router's advertisement interval and a MN can therefore arrive on a link at any time during the interval. On average, the MN will arrive on a link halfway between two consecutive advertisements and will have to wait for half the advertisement interval before the first new advertisement is received.

The worst-case scenario occurs when a MN arrives immediately after an advertisement is broadcast. When using the Neighbour Discovery minimum value of 3–4 seconds, a MN will wait 1.75 seconds on average before the first new advertisement is received (assuming a mean interval of 3.5 seconds). In the worst-case, a MN will experience a delay of 4 seconds. When using Mobile IPv6 minimum values (0.03–0.07 seconds), the average and worst-case results are reduced to 25 ms and 70 ms respectively. Equation 3.1 defines the average time until a new advertisement is received after arriving on a link (AGENTDISCOVERY) where the advertisement interval varies randomly between MAXINTERVAL and MININTERVAL[8]. When an eager-binding policy is used (and the new advertisement is unheard), Equation 3.1 represents the time taken to perform movement detection.

$$\text{AgentDiscovery} = \frac{(\text{MaxInterval} + \text{MinInterval})}{4} \tag{3.1}$$

Both the Mobile IPv4 and Mobile IPv6 lazy-binding policies rely on missed advertisements as a sign of network movement. If it is assumed that a MN leaves its previous network halfway through the mean advertisement interval, then the average time until the advertisement lifetime expires and movement is detected (RESIDUALLIFETIME) is expressed by the following formula:

$$\text{ResidualLifetime} = (0.75 + (n - 1)) \times \text{MaxInterval} - 0.25 \times \text{MinInterval} \tag{3.2}$$

where n is the number of missed advertisements that the MN considers an indication of movement [20]. When movement is detected after only one interval and Neighbour Discovery minimum rates are used, the MN will detect movement after 2.25 seconds (on average). For Mobile IPv6 minimum rates, the expected time before the interval expires is 45 ms.

3.6.2 Link Layer Hints

Many movement detection optimisations aim to bypass the need for frequent periodic advertisements by decreasing the layer independence between Mobile IP and the link layer. A communication channel is established between these two layers, allowing control information to be passed from the link layer to Mobile IP. This information primarily assists

[8]For a derivation of the following equations, refer to Appendix C.

the Mobile IP layer in performing faster movement detection. In addition, link layer information makes movement detection less dependent on (or even independent from) periodic advertisements. Lower advertisement rates can be employed which improve the overall bandwidth efficiency. Information that is passed from the link layer to Mobile IP is termed a link layer *hint* or *trigger*. Although some publications specify a difference between these two terms, they will be used synonymously in this study[9].

In many cases, a link layer hint does not conclusively determine that network layer movement has occurred. For example, a simple link layer hint would be an indication to a MN's Mobile IP layer that a link layer handoff has just occurred. This hint may occur as a result of layer 2 or layer 3 handoff, and this ambiguity should be taken into consideration.

Several different types of hints have been proposed in the literature. The next chapter will investigate these more closely and verify whether they can be applied to 802.11 wireless LANs. Some techniques that use link layer hints to optimise movement detection will be described in the following sections. Many of these optimisations are presented within the context of Mobile IPv6. It should be borne in mind that these proposals are equally relevant to Mobile IPv4.

3.6.3 Hinted Cell Switching

The Hinted Cell Switching (HCS) technique has been proposed by Fikouras and Gorg [36]. A MN's Mobile IP layer is made aware of link layer events and receives a simple indication when a link layer handoff has occurred. Once this indication is received, the MN requests a router advertisement by broadcasting a solicitation. All access routers on the (new) link reply with their advertisements, including all necessary options such as advertisement interval, prefix option etc. This information allows the MN to detect movement and configure itself on the new network. Figure 3.9 illustrates the link layer hint graphically while the diagram that follows (Figure 3.10) depicts the message sequence of HCS movement detection.

There are three main drawbacks to HCS. Firstly, after every link layer handoff, HCS generates a significant amount of broadcast traffic (RS/RA messages) which, as stated earlier, is undesirable. A link layer handoff also does not necessarily imply network level movement. This can be seen clearly in Figure 3.2 where a MN performs a link layer handoff

[9] A "trigger" is sometimes used to denote a layer 2 event indication whereas additional layer 2 information is termed a "hint" [26]

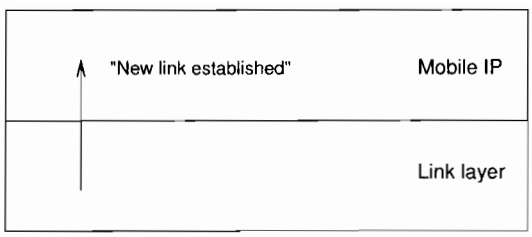

Figure 3.9: Link layer hint for HCS

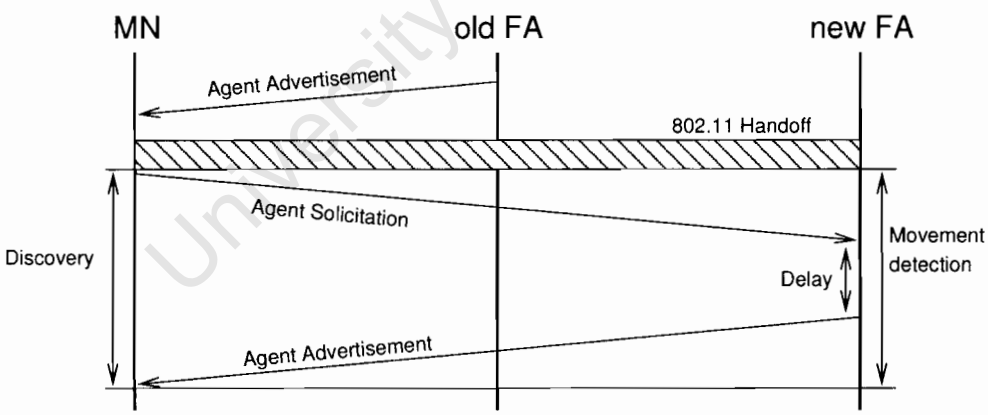

Figure 3.10: HCS message sequence

between two APs within the same IP network. As a result, in most cases this RS/RA traffic is unnecessary because no IP movement has occurred.

The second problem with this technique is due to the manner in which broadcast packets are dealt with in 802.11 wireless LANs. The IEEE 802.11 standard [46] considers broadcast and multicast packets to be asynchronous in nature. "Due to the characteristics of the WM [wireless medium], broadcast and multicast MSDUs [MAC service data units] may experience a lower quality of service, compared to that of unicast MSDUs" [46]. As a result, these broadcast/multicast messages may be reordered to improve the reliability of other unicast packets. In fact, during preliminary tests, it was found that the commercial APs under investigation do not forward broadcast[10] advertisements over the wireless medium at all.

The third drawback of HCS arises from the delays built into a solicitation/advertisement exchange, as specified by IPv6 Neighbour Discovery and IPv4 ICMP Router Discovery. Recall that in a Mobile IPv6 network, a router solicitation should be delayed up to 1 second and a replied advertisement must be delayed up to 500 ms to prevent synchronisation. With these values, the average and worst-case delays until a solicited advertisement is received are 750 ms and 1.5 seconds respectively. These delays contribute significantly to the total disruption caused by movement detection [21].

One way of eliminating the initial solicitation delay is to include it only under certain conditions when it will not disrupt a MN's traffic. Solicitation delays should be employed in a solicitation/advertisement exchange under the following conditions [20]:

- The MN has no upper-layer sessions.
- The MN has no sessions that have sent or received data for a certain amount of time (120 seconds).
- The MN has alternative interfaces with valid CoAs that are handling all session traffic.

Under these conditions, the average and worst-case delays are reduced to 250 ms and 500 ms respectively

[10] with destination address 255.255.255.255

3.6.4 Fast Router Advertisement

An additional technique, used in conjunction with HCS, has been proposed that removes the random delay between a router solicitation message and its replied advertisement [52]. This mechanism, known as Fast Router Advertisement (FastRA), is not only applicable to HCS, but to any mechanism relying on solicited advertisements. FastRA allows a specific router to respond immediately to a solicitation with a router advertisement. The delay is removed from only one router on an IP subnet, ensuring that solicitations are not replied to simultaneously by several routers. Furthermore, a FastRA router will only transmit a certain number of undelayed RAs within a specific interval. Additional solicitations that arrive after this maximum number is reached are discarded until the next unsolicited RA. Figure 3.11 shows a simple timing diagram of a normal RS/RA exchange (a) and the improvements due to FastRA (b).

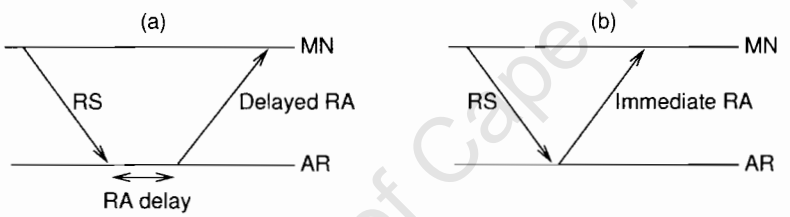

Figure 3.11: Normal RS/RA sequence (a) FastRA sequence (b)

Thus far, an automatic mechanism that selects which router on an IP subnet will use FastRA has not been defined, other than through external administration. The Deterministic FastRA protocol extends FastRA and allows the routers in a subnet to negotiate the sequence in which all routers will answer solicitations [22]. Each router in the subnet is allocated a ranking that defines how quickly it will respond to solicitations. As described previously, only one FastRA router will answer solicitations immediately. However, Deterministic FastRA delegates a replacement FastRA router when an existing FastRA router malfunctions or goes off-line. For further information, refer to the Deterministic FastRA Internet draft [22].

3.6.5 Fast Hinted Cell Switching

The Fast Hinted Cell Switching (FHCS) mechanism [37] was developed to avoid the need for broadcasting RS/RA messages after a link layer handoff. FHCS extends the information communicated from the link layer to Mobile IP. A FHCS hint not only includes an indication that a link layer handoff has occurred, but also includes IP information in the hint. For example, the hint can contain the identity of the local mobility agent/access router along with additional IP configuration information. How these hints are generated and what type of information they convey will be discussed in greater detail in the next chapter. Figure 3.12 illustrates the types of parameters that constitute a FHCS hint.

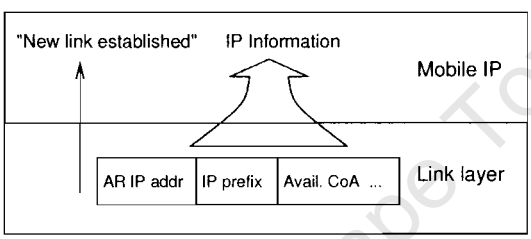

Figure 3.12: Link layer hint for FHCS

In FHCS, a MN's link layer is able to gather IP information about the current network after layer 2 handoff completion. This information is immediately provided to the Mobile IP layer in the form of a hint (Figure 3.12). Ideally, FHCS allows the movement detection process to be avoided completely because a MN is no-longer restricted to receiving IP information via an advertisement. This information can be provided directly by its link layer. However, in practice this technique faces a number of obstacles, such as how the link layer should acquire network layer information. These issues are also investigated further in the next chapter.

3.6.6 Advertisement Caching

The Advertisement caching scheme, known as Fast Router Discovery (FRD) [16, 17], also allows a MN to perform movement detection without having to resort to solicitation/advertisement exchanges. In this technique, router advertisements are delivered to a MN as soon as it attaches to the IP subnet. This can be achieved by incorporating

additional network layer functionality into 802.11 access points in addition to their normal layer 2 bridging operation. An AP is thus able to cache the most recent advertisements broadcast by on-link routers and deliver them to newly associated MNs. Figure 3.13 depicts the performance of advertisement caching.

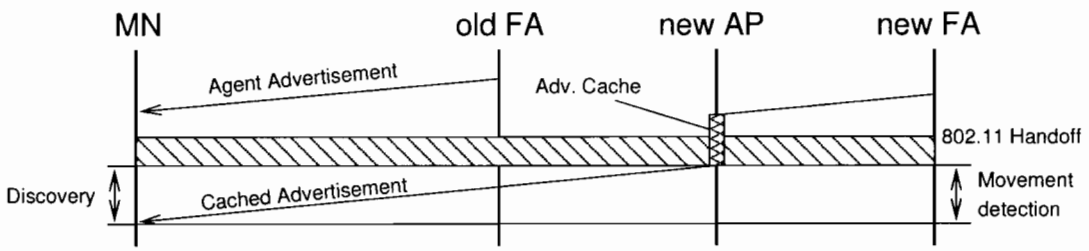

Figure 3.13: Advertisement caching message sequence

This mechanism is somewhat different to the previous optimisations. In all the above proposals, the MN receives hints from its link layer and acts accordingly. In this proposal, the AP receives a hint that a new station has been connected. Once this hint is received, the AP delivers the most recent RA to the MN. There are variations to this technique where the advertisement caching functionality is moved from the APs to a separate entity. These details will be investigated further in the next chapter.

One of the advantages of this movement detection scheme is that it limits much of the broadcast RA traffic to the BSS of an AP instead of the whole IP subnet. Routers can use much lower advertisement rates and rely on APs to deliver RAs timeously. This mechanism also has the potential to provide the fastest movement detection performance [20]. This can be seen clearly in Figure 3.13.

3.6.7 Hybrid Technique

Most of the movement detection optimisations that have been discussed thus far have focused on speeding up the delivery of the first advertisement to a MN after it connects to a new IP network. This represents the agent discovery stage of movement detection. All of these techniques imply that an eager-binding selection policy will be used to perform a Mobile IP handoff after the first new advertisement is received [29].

An eager-binding policy is very useful when a MN performs hard Mobile IP handoffs

between 802.11 networks. The reason for this is that after a MN establishes an 802.11 link to a new IP network, old routers/agents are no longer reachable and only local routers/agents can be detected. However, while a MN is attached to an IP subnet, an eager-binding policy can lead to unnecessary Mobile IP handoffs between mobility agents on the same network. This can even occur without physical terminal movement, as a result of receiving an agent advertisement that has not been heard for a period of time (classified as "unheard"). The MN will incorrectly interpret this "unheard" advertisement as an indication to register with the "new" agent. For example, an agent may not broadcast advertisements for some time (e.g. due to malfunction). As a result, all eager-binding MNs on the network will immediately attempt to register through the agent when it eventually does advertise its presence. When large numbers of nodes exhibit this behaviour, the network may be flooded with registration requests for some time. These requests may also overwhelm the "new" agent.

Another example of this inefficiency occurs as a result of packet loss. If a MN misses a certain number of advertisements from an additional local agent (other than the currently registered agent), the MN removes its entry from its list of heard agents[11] [28]. 802.11 nodes are relatively prone to missing advertisements for several reasons. Firstly, broadcast advertisements may be assigned a lower priority than unicast messages. Secondly, 802.11 handoffs last a significant time, during which no packets may be received. Furthermore, a MN may be associated to an AP that blocks broadcast advertisements (255.255.255.255) but forwards multicast advertisements (224.0.0.1). The MN will therefore be unable to passively detect the agents using broadcast advertisements. Lastly, 802.11 wireless LANs usually have little or no network planning, and a MN may temporarily move into a region with little or no radio coverage as a result. If local agents are using different advertisement broadcast rates, the MN will remove some entries from the list of heard agents before others. When the MN begins receiving advertisements from these "unheard" agents, registration will be attempted even if the current agent is still valid.

A Mobile IPv6 node using an eager-binding may perform unnecessary configuration procedures when receiving advertisements from several on-link access routers [60]. Several studies have investigated the eager-binding policy, both theoretically and through emulations. Many of these have also highlighted the problems listed above, along with further security issues that have not been considered in this study [60, 11, 17, 20].

These issues motivated the author to develop a hybrid technique that combines several

[11] Even if the MN remains attached to the same IP subnet.

movement detection optimisations together. The essential principle behind this technique is that a lazy-binding should be used while a MN is connected to an IP subnet (to prevent unnecessary IP handoff signalling) and that an eager-binding should be used for inter-subnet handoffs. In the hybrid system, detailed link layer information is available to the Mobile IP layer which assists in performing accelerated but stable movement detection. Because it is so dependent on link layer information, this system will be discussed within the context of 802.11 wireless LANs.

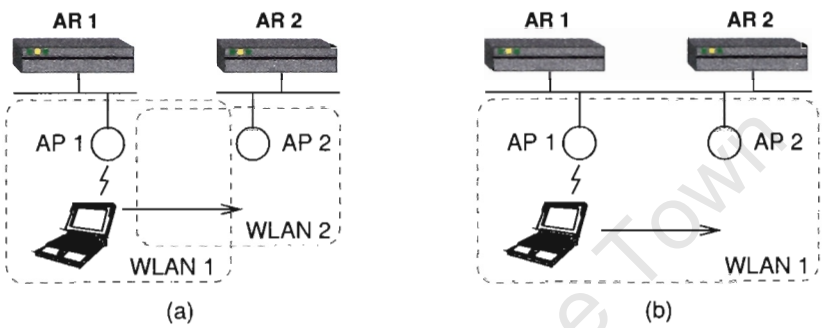

Figure 3.14: Inter-subnet (a) and Intra-subnet movement (b)

The hybrid movement detection scheme should result in robust performance for the different scenarios illustrated in Figure 3.14. Figure 3.14 (a) depicts a MN moving from one wireless LAN on one subnet to another wireless LAN on a neighbouring subnet. In this case, movement detection must be performed as quickly as possible. However, (b) represents intra-subnet terminal movement. In this case the MN should avoid performing a Mobile IP handoff even after receiving advertisements from different ARs, or after performing an 802.11 handoff within the subnet.

The hybrid technique relies on one of the previous movement detection optimisations to ensure that a MN receives an advertisement as soon as it attaches to a new network (Figure 3.14 (a)). In addition, link layer parameters, such as signal-to-noise ratio (SNR), are used to determine what selection policy should be used. These parameters are used to roughly predict when an 802.11 handoff might occur. Thus, when a MN is well within range of its access point, a lazy-binding selection policy is activated. If several additional routers reside on the link, the MN will ignore their advertisements as long as the current router is reachable. However, as the MN moves further away from its AP and closer towards an AP on a neighbouring network, the link signal quality will degrade. When the link quality

passes below a certain threshold, an eager-binding policy will be activated in anticipation of possible network movement.

The above description assumes that terminal movement always affects the 802.11 link quality, thereby initiating link layer handoffs. However, external signals from a user, device driver or operating system may also initiate an 802.11 handoff. These signals serve as additional indications that a link layer handoff is about to occur and cause the eager-binding policy to be activated.

A further improvement can be included in the hybrid system. As can be seen from Figure 3.14 (b), not all link layer handoffs result in network movement. Ideally, a MN should not use an eager-binding when moving between AP 1 and AP 2. Therefore, the MN will maintain a list of the link layer (MAC) addresses of all its surrounding 802.11 APs. This list will also identify which APs reside on the local network. A MN can build such a list by accessing information gathered in the scanning stage of a 802.11 handoff or through a mechanism such as IAPP. Once the MN has completed the link layer handoff, the AP list will be used to determine if the new AP resides on the current IP network. If it does, then the eager-binding is deactivated immediately. Possible mechanisms that determine whether or not an AP is part of the local subnet will be discussed in the next chapter.

Chapter 4

Movement Detection Optimisation Design

4.1 Introduction

A general architecture that aims to minimise Mobile IP handoff latency through improved movement detection and micromobility has been outlined in the previous chapter. Motivations for several of these movement detection enhancements have been described. The different concepts that form the basis of these proposals will now be developed into a more detailed design. This chapter focuses on the software architecture of each movement detection optimisation. A number of aspects pertaining to these different techniques will be considered and compared. For example, many schemes can be implemented using a variety of alternative designs. It will be shown that although some of these designs (theoretically) support near-instantaneous movement detection, they also pose significant practical obstacles. The main aim of this chapter is to specify mechanisms that allow these optimisations to incorporate the facilities of 802.11 technology. The practicality of these different designs and their effectiveness in accommodating VoIP systems will also be investigated.

As stated earlier, seamless handoff represents the goal of any Mobile IP handoff technique. Seamless handoffs are currently unattainable within the architecture under investigation due to the delays that 802.11 handoffs and generic movement detection introduce. Smooth handoffs may be achieved by using large buffers or by some tunnelling mechanism such as fast/low-latency Mobile IP [31, 29]. However, when VoIP is the application under consideration, large buffering schemes are difficult to implement because they introduce

unacceptable delays [50]. In addition, fast/low-latency techniques cannot be used effectively in infrastructure 802.11 wireless LANs due to the constraints that 802.11 places on Mobile IP [18, 70].

Therefore, the specific aim of any design presented below is to allow low-latency movement detection to be performed. This will result in a reduction of both the total Mobile IP handoff latency and the associated packet loss. Ideally, if the delays from other handoff stages are minimised, these new mechanisms should reduce movement detection delays to the point where a VoIP connection is not significantly affected by Mobile IP handoffs. This will be verified through experimentation, as discussed in later chapters.

Although many of the designs discussed below have been proposed in the literature, this chapter will develop them further by including several extensions. These designs will only be discussed within a Mobile IPv4 framework in order to simplify any discussion that ensues[1].

4.2 802.11 Link Layer Hints

Before any 802.11 link layer hints or movement detection optimisations are discussed, it is important to highlight the effects that 802.11 technology have on movement detection in general. These effects are revised below:

- 802.11 forces Mobile IP to perform hard handoffs. An 802.11-enabled MN can only detect agents on its current link and cannot receive advertisements from neighbouring agents.
- Even if a MN's Mobile IP layer detects the completion of link layer handoff immediately, it cannot determine agent/network information without an advertisement.
- 802.11 handoff decisions are usually made unanimously by the hardware/firmware of a particular device. Anticipation of 802.11 handoffs is therefore extremely difficult in current commercial devices because link layer processes are completely hidden from upper-layers. Upper-layers are also unable to specify the AP that a wireless card should handoff to.

[1]Many of the original proposals from the literature were based on Mobile IPv6 and these have been suitably adapted to Mobile IPv4.

- When an 802.11 handoff is performed, AP selection is only finalised towards the end of the link layer handoff process[2]. In addition, the wireless medium is highly volatile. As a result, even when a list of candidate APs is available, upper-layers are unable to accurately predict where an 802.11 device will handoff to.
- The 802.11 standard assigns a lower priority to broadcast or multicast packets in order to reliably transport unicast messages. This is further worsened when dynamic rate scaling is used. When lower data rates are activated, broadcast/multicast packets may be delayed to an even greater extent.

As a result of these factors, Mobile IP is always forced to perform movement detection in a reactive manner, where physical network movement is discovered after a new link is established. Movement detection therefore has to rely on IP advertisements[3]. The previous chapter illustrated that this can be improved by allowing link layer hints to influence the movement detection process. A closer look is now taken at the different kinds of hints that 802.11 can provide.

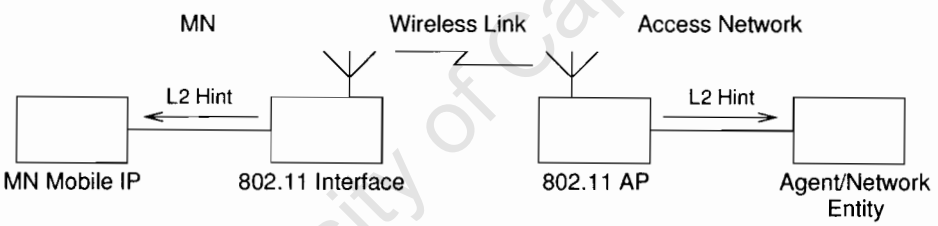

Figure 4.1: Link layer hint network model

The model depicted in Figure 4.1 [79] illustrates the entities that can generate and receive link layer hints. Hints can be generated at the MN or by the network. On the MN side, hints are generated by the MN's wireless network interface and communicated to upper-layers. Likewise, hints are generated on the network side by 802.11 APs. These hints may be communicated to another entity, such as a foreign agent, using a layer 2 or layer 3 protocol.

The two simplest hints that the 802.11 layer can provide are indications that a wireless link has just been established or removed. These hints are termed *link-up* and *link-down*

[2]Recall that the scan phase takes up to 90% of an 802.11 handoff delay [57].

[3]Unlike 3GPP and 3GPP2 technologies, 802.11 networks are unable to automatically provide IP configuration information with a link-up hint [26].

hints respectively. A link-up hint is generated when the MN's 802.11 wireless interface receives a positive association or reassociation response message from its AP. These messages are used because they signal the end of an 802.11 handoff. Similarly, the reception of a deauthentication or a disassociation message signals that the MN no-longer has a valid connection to the network. A link-down hint is therefore induced.

Unfortunately, the reception of association messages is usually very difficult to detect by upper-layers. However, several more indirect methods of detecting 802.11 handoffs exist. While a MN is associated with an AP, the current ESSID is usually easily accessible to upper-layers and can be used to generate hints. Until now it has been assumed that an IP subnet will include only one wireless LAN using a single ESSID. This means that a one-to-one relationship exists between an ESSID and an IP network. Under these conditions, the ESSID can be monitored and any change in ESSID would translate to network movement. An ESSID change would therefore serve as a relatively reliable hint to Mobile IP that the MN is on a new network and that an IP handoff must be performed.

However, the assumption of a one-to-one mapping between ESSID and IP network is usually not very realistic. In practice, large wireless LANs using a single ESSID may provide access to several IP subnets (e.g. within an administrative domain). Conversely, a single IP network can contain several wireless LANs, each using a different ESSID. Therefore, observing an unchanged ESSID does not directly guarantee that the MN is still on the same IP network. However, a change in ESSID can be interpreted to mean that network movement *may* have taken place [26, 25]. This technique is made even more unreliable by the fact that ESSIDs are non-unique identifiers (similar to a "network name"). Any identifier can be used as an ESSID and often default factory settings are used. This makes it extremely difficult to derive a relationship between an ESSID and the IP network it resides on. As a result, if this technique is used, all hints should be considered weak and should be confirmed with additional information.

Another way of notifying Mobile IP that a link layer has occurred is by detecting and reporting a change in the associated AP's identity. This can be achieved by monitoring the current AP's link layer MAC address. A change in the current AP's MAC address implies that the MN has associated with a different AP. This allows a hint to be generated, informing the Mobile IP layer that a link layer handoff has occurred.

The link-up/down hints introduced above are termed *event* triggers because they indicate that the link layer state has changed. On the other hand, *predictive* hints signal that

a link layer change may happen in the future [43]. Three predictive hints have been defined (termed source, target and MN triggers) that occur before a link layer handoff takes place [24, 34]. The details of these triggers will not be discussed as they cannot be implemented in 802.11 networks [18, 70]. The reason for this is that link layer handoff, along with the identities of the new (target) AP/FA, cannot be accurately predicted.

A predictive trigger can also be initiated using terminal location information to predict a layer 2 handoff [69, 14]. When location information is available to a MN, it can be used to indicate that a link layer handoff is imminent [69]. These triggers can also be achieved, even when considering technologies that do not include support for location information, by making use of GPS information [14]. However, these techniques are not considered further because 802.11 does not inherently support location information and specialised hardware would be required, such as GPS receiver.

4.3 Hinted Cell Switching

4.3.1 Link Layer Hints

The link layer hint that initiates HCS is both generated and received within the MN. The hint is generated immediately after a link layer handoff has been completed. This movement detection scheme makes use of simple link-up hints. These can be generated in one of two ways. The most effective mechanism for providing link-up hints would be to modify the wireless card's device driver. The driver would include additional functionality allowing it to signal the occurrence of a link layer handoff to upper-layer Mobile IP entities (as depicted in Figure 4.2). This could be done using an event, interrupt or some other form of interprocess communication.

This hint generation technique is both immediate and efficient in that link layer information is passed directly up to Mobile IP as it becomes available. It is also relatively easy for a driver to use association response messages to signal these hints. However, in order to implement these changes, individual drivers must be modified. This was found to be a complicated procedure, as device drivers are difficult to modify and debug effectively. This solution is also not portable to devices from other manufacturers (using different chip-sets) as different drivers are needed.

Figure 4.3 illustrates an alternative mechanism for generating link layer hints. Instead

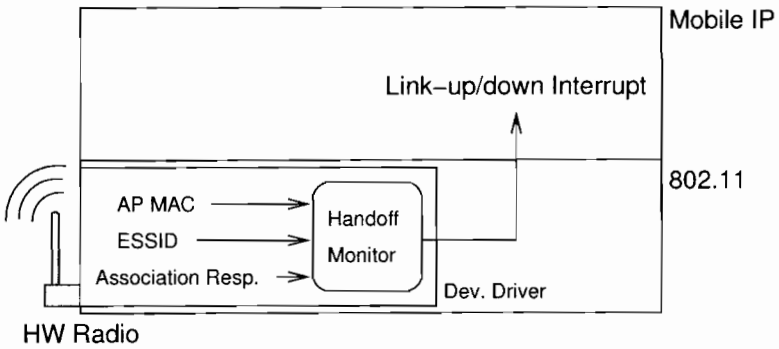

Figure 4.2: Hints generated by device driver

of including additional functionality within a device driver, a separate external entity continuously monitors information available from the driver. This monitoring entity will not be able to detect association messages easily because it does not operate within the link layer. The current ESSID, AP MAC address or other operating system information is therefore probed to determine when a layer 2 handoff has occurred. Once a handoff takes place, the monitor passes a link-up hint to the Mobile IP layer. The monitor may be included as part of the Mobile IP layer or may be executed as a separate application-layer process.

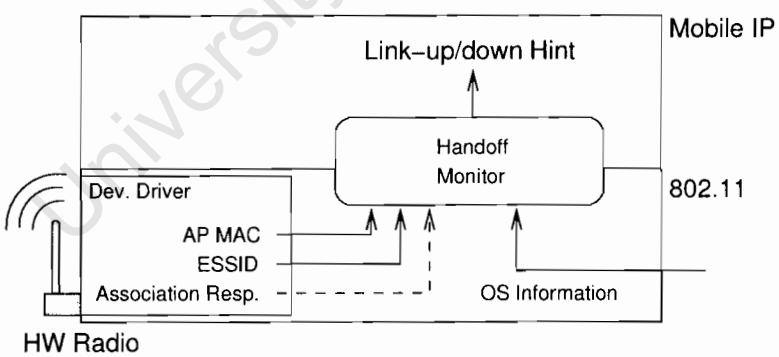

Figure 4.3: Hints generated by external monitor

Most 802.11 drivers provide simple access to ESSID and AP MAC address information. For example, under the Linux operating system, the "wireless-tools" `iwlib` library allows this information to be retrieved by user applications. However, the monitoring entity may

also benefit from device-specific facilities. For example, the Agere Systems Orinoco devices used in this study report certain link layer events (such as association and disassociation) to the main Linux operating system log file. By monitoring these log messages, association and disassociation events can be detected by the monitor and passed to Mobile IP as a link-up or link-down hint.

Initially, this monitor mechanism raised several concerns. Firstly, hint generation is not as efficient as when modified drivers are used. Hints are not generated immediately by the monitor and the speed at which events are reported depends on the polling rate. System resources are therefore wasted because the monitor must frequently poll the device driver in order to ensure that hints are generated within a short space of time. In addition, reliance on device-specific facilities is usually implemented using non-standard mechanisms as opposed to a general well-defined interface. This can give rise to portability issues. Despite these concerns, a practical investigation into a similar mechanism found that the AP MAC address can be probed using a system timer routine executed every 10 ms. Each probe verifies the identity of the AP using only a few CPU clock cycles [70]. As a result, this technique offers a simple and adequate way of generating link-up hints.

4.3.2 Functional Design

The finite state machine for the HCS algorithm is shown in Figure 4.4 (adapted from its original [37]). A MN using HCS can be in one of four states: IDLE, LINK DISRUPTION, AGENT SELECTION and REGISTRATION. Initially, the MN begins in the IDLE state. If the advertisement lifetime of the current mobility agent expires, then the AGENT SELECTION state is entered directly (similarly to the lazy-binding scheme). However, the reception of a link-up hint causes the MN to enter the LINK DISRUPTION state. In this state, the MN broadcasts an agent solicitation and waits for a replied advertisement. If the replied advertisement is from its current agent, then the MN returns to the IDLE state. However, if an advertisement from a previously unheard mobility agent is received, then the MN enters the AGENT SELECTION state. In the AGENT SELECTION state, the MN selects the new agent and configures its new CoA. This CoA is then registered (REGISTRATION) and if a positive registration reply is received, the MN returns to the IDLE state. If a negative registration response is received, the MN returns to AGENT SELECTION and chooses an alternate agent (if available).

The eager-binding policy is implemented in the agent selection state. This is clear from

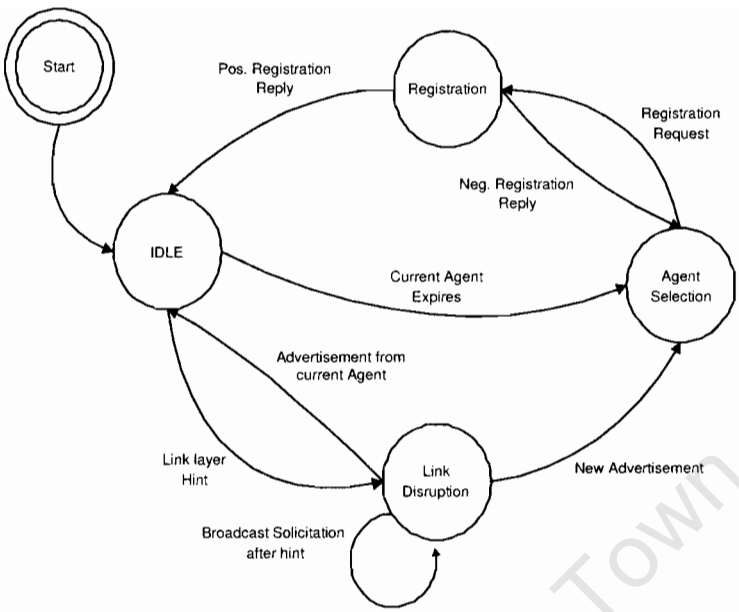

Figure 4.4: HCS finite state machine

the description above, because AGENT SELECTION immediately initiates REGISTRATION if an unheard advertisement is received.

In order to maximise HCS performance, foreign agents should incorporate FastRA (or FastADV[4]) functionality to allow solicitation/advertisement exchanges to take place as quickly as possible. This minimises the time spent in the LINK DISRUPTION state, indicated in Figure 4.4 above.

4.4 Fast Hinted Cell Switching

4.4.1 Link Layer Hints

Fast hinted cell switching (FHCS) also relies on event hints after an 802.11 handoff has been completed. However, additional IP information is included in these hints that allows the MN to configure its CoA without any solicitation/advertisement exchanges. The hint provides the MN's Mobile IP layer with the identity of the local mobility agent. Registration

[4]FastADV (fast advertisement) is the author's MIPv4 adaptation of the MIPv6 FastRA proposal.

can therefore be initiated immediately after link layer handoff.

There are different approaches that allow layer 2 entities to carry IP information. The simplest way of achieving this is by setting the ESSID of a wireless LAN to the IP address of the on-link mobility agent. Whenever the MN detects a change in ESSID, using either a monitoring entity or driver interrupt, the new ESSID will be used as a foreign agent CoA [37]. If it is assumed that only one mobility agent operates per subnet, a one-to-one relationship exists between the ESSID and IP subnet. ESSID changes can therefore be used as a reliable hint of network movement (Figure 4.5).

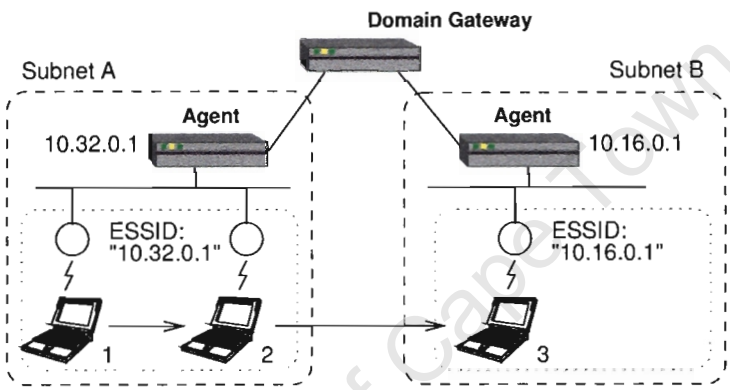

Figure 4.5: ESSID used to transport IP information

A more elaborate solution has been proposed that extends IEEE 802.11 management frames to include a new application-specific information element [75]. 802.11 management frames include probe request/response, beacon, authentication and association frames [46]. These frames are composed of fixed fields and information elements (IE). Information elements are used to carry variable length or optional information and have a well defined structure, similar to IP extension headers. This facilitates the definition of new IEs that will contain optional IP information. The following design is based on this proposal.

Application-specific IEs can transport a variety of IP parameters. An AP would typically include the IP network prefix (netmask) along with the on-link mobility agent's MAC/IP address. This allows the MN to use the agent's IP address as a FA CoA (or determine if it is on the home network). Prefix information can also be used to confirm that network movement has occurred.

The advantage of using a separate application-specific information element instead of the

Subnet Prefix	ESSID	AP-MAC	Signal Quality	CoA	Status
prefix1	essid1	MAC1	x dBm	agent_CoA1	-
prefix1	essid1	MAC2	x dBm	agent_CoA1	active
prefix2	essid2	MAC3	x dBm	agent_CoA2	-

Table 4.1: MN 802.11 link layer scan results

ESSID field to contain IP information is that the ESSID field is no longer restricted. The correlation between an ESSID and IP network is removed, and the same ESSID can be used across an administrative domain and can be set to any desired identifier. This design is also scalable, as further extensions can be defined to include additional IP information in link layer frames.

When a MN's 802.11 interface performs a scan of surrounding APs, a list of received beacon information is compiled, as shown in Table 4.1. This list contains both 802.11 and layer 3 information included in application-specific IEs. The scan can be executed as part of a layer 2 handoff or while the device has available bandwidth (e.g. when idle). For example, while the MN in Figure 4.5 is in position 2, an 802.11 scan produces the list of adjacent APs (including those on different networks and using different ESSIDs) as described by Table 4.1.

When a link layer handoff has been completed, the MN uses the new AP's identity (MAC address) or ESSID to immediately establish the IP address of the on-link agent and configure its CoA. This mechanism makes use of a combination of event and predictive hints. Before an 802.11 handoff occurs, the list of surrounding APs and their corresponding mobility agent information is formed that predicts the possible CoA settings that will be needed in the event of any layer 2 handoff. Once a layer 2 handoff occurs, an event hint containing the new AP's MAC address notifies the Mobile IP layer that the appropriate agent should be selected using the AP/agent list.

The above techniques allow movement detection to be bypassed completely because the on-link mobility agent's IP information is immediately available when a new 802.11 link is established. This also allows a new CoA to be configured automatically (using the FA CoA mode). However, these techniques are not without drawbacks. Firstly, no automated mechanism has been specified that allows layer 3 information to be disseminated to 802.11 APs, allowing them to set their ESSID or application-specific IEs appropriately. One possi-

ble way of achieving this is by allowing APs to monitor agent advertisement broadcasts and use their included information. Secondly, the addition of new extensions to 802.11 Management frames is very difficult to implement using commercial products because their 802.11 functions are coded into the hardware/firmware. Lastly, these techniques involve serious security issues that must be considered. For example, in a wireless environment, it is easy for an attacker to broadcast false IP information that could mislead MNs into incorrectly configuring themselves. Although these security problems are of great significance, they do not fall within the scope of this study.

4.4.2 Functional Design

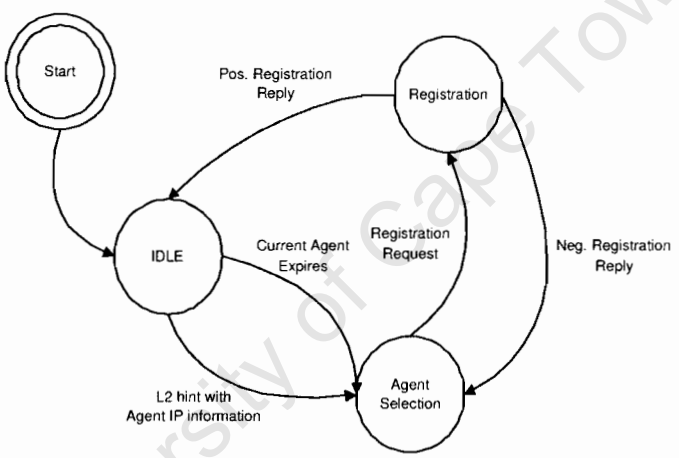

Figure 4.6: FHCS finite state machine

The finite state machine for the FHCS algorithm, shown in Figure 4.4 [37], consists of only three states: IDLE, AGENT SELECTION and REGISTRATION. During the idle state, the MN collects 802.11 scan information and builds its list of neighbouring APs. The LINK DISRUPTION state is not applicable in this case because a link layer hint allows the MN to move directly from IDLE to AGENT SELECTION, using the information contained in the hint. Once the MN's Mobile IP layer selects the new agent, the REGISTRATION state is entered as stated above. When registration ends successfully, the MN returns to the IDLE state.

4.5 Advertisement Caching

4.5.1 Link Layer Hints

This movement detection optimisation relies on the reception of a link layer hint by an entity on the access network. The entity delivers a cached advertisement to the MN as soon as a valid link is established to the network. There are two different approaches that can be adopted in order to facilitate these network hints.

The first network entity that detects a newly attached MN is, by definition, an 802.11 AP. This occurs after an AP transmits a positive association response to the MN. At this point, a new logical connection between the MN and the rest of the network is established by the AP. Therefore, link layer hints generated within the access network always originate from 802.11 APs.

In the first solution (introduced in the previous chapter), 802.11 APs perform the caching and delivery functions [17]. An AP scans[5] all incoming layer 2 frames that will be bridged from its wired interface to the wireless medium. When it detects an unsolicited broadcast agent advertisement, the AP saves a local copy of the advertisement. An AP may either scan every incoming layer 2 frame or scan frames during periodic intervals. Alternatively, an AP with additional built-in IP functionality can request an agent advertisement by transmitting an agent solicitation.

When a new MN requests an association with the AP, the AP replies with an association response message that includes the cached advertisement. The cached advertisement information can be transported by a positive association response within an additional information element (as discussed above). When the MN receives the association response, a link-up hint is generated that includes the advertisement information. As can be seen from Figure 4.7, this mechanism requires both the MN and APs to support 802.11 link layer hints.

The alternative mechanism is significantly more flexible and is based on a combination of several proposals [70, 66]. In this case, the advertisement caching functionality is moved to a separate *caching agent*. The caching agent resides on the wired section of an IP network and caches agent advertisements. This entity also performs two additional functions. Firstly, agent solicitations broadcast by MNs are detected by the caching agent, and a cached

[5]Note the the term "scan" used here bears no relation to an 802.11 scan (using probe requests/responses).

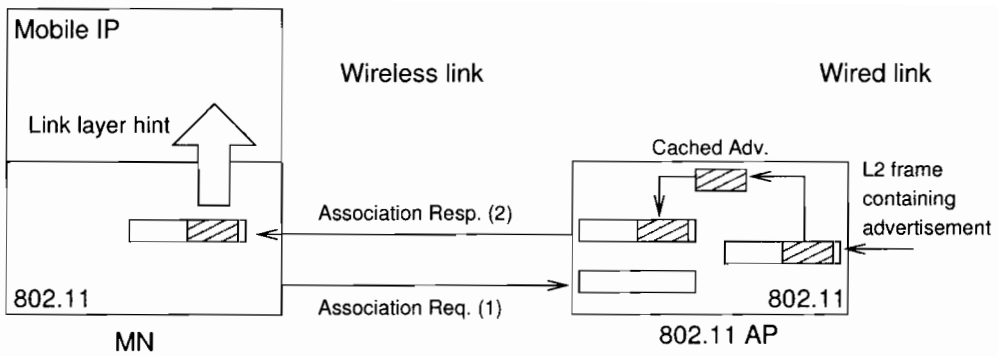

Figure 4.7: Advertisement caching performed by 802.11 AP

advertisement is transmitted immediately in response. This behaviour is based on the same principles as the FastRA mechanism discussed in the previous chapter.

Instead of relying on broadcast solicitations and advertisements, a more elegant solution would be for the caching agent to receive a unicast solicitation and to reply with a unicast advertisement. In order to accomplish this, the caching agent associates itself with a well-known, faked MAC address (e.g. `a:b:c:d:e:f`) [70]. The caching agent periodically broadcasts dummy layer 2 frames using this address as a source address. These messages ensure that all network nodes (including hosts, switches and APs) update their link layer forwarding tables so that frames addressed to the faked MAC address are directed to the caching agent. Furthermore, all MNs are assumed to know this address. When a MN receives a link-up trigger after a link layer handoff, an agent solicitation is sent to the unicast `a:b:c:d:e:f` address. The caching agent then responds with a cached unicast advertisement to the MN's MAC address. The term "unicast" in this context refers to the layer 2 destination address and not to the IP address (as it usually does). In this situation, it is unimportant that the MN's IP addresses may be invalid on the current subnet or that broadcast IP addresses are used in advertisement/solicitation packets. The reason for this is that when hosts reside on the same link, a link layer protocol can be relied upon to deliver these messages.

The second function of a caching agent is to deliver a cached, unsolicited advertisement to any newly associated MNs. An 802.11 AP generates a link layer hint when a MN is successfully associated, and this hint is forwarded to the caching agent[6]. When the

[6]The exact protocol for transporting this hint between the AP and the caching agent will not be

caching agent receives the hint, the cached advertisement is broadcast over the link. If the AP includes the MN's MAC address in the hint, the caching agent can send a unicast advertisement directly to the MN using its MAC address.

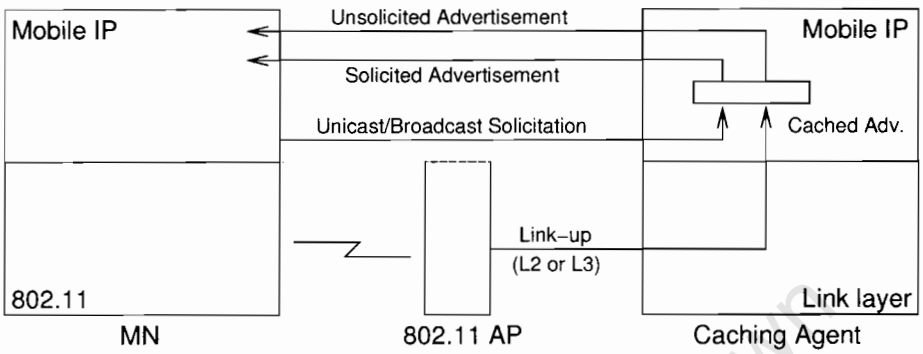

Figure 4.8: Advertisement caching performed by a separate caching agent

Figure 4.8 illustrates the different message exchanges that a caching agent supports. A MN can use either a unicast or broadcast solicitation/advertisement exchange.

4.5.2 Functional Design

The advertisement caching technique using 802.11 APs also reduces the finite state machine to three states (i.e. all broken states and transitions in Figure 4.9 are ignored). This finite state machine is essentially the same as for the FHCS technique. The only difference is that a new advertisement causes the MN to enter the agent selection state.

The finite state diagram in Figure 4.9 illustrates the redundancy that is built into the caching agent mechanism. If the MN misses the initial cached advertisement that is sent immediately after the link is established, it *may* solicit the caching agent. In this case, the MN enters the LINK DISRUPTION state where it waits for a reply. The caching agent immediately responds to the solicitation with a cached advertisement. This brings the MN into the AGENT SELECTION state. Using a unicast solicitation and advertisement exchange is advantageous in 802.11 wireless LANs because these messages, unlike their broadcast counterparts, are not assigned a lower priority by 802.11 entities.

investigated in this study.

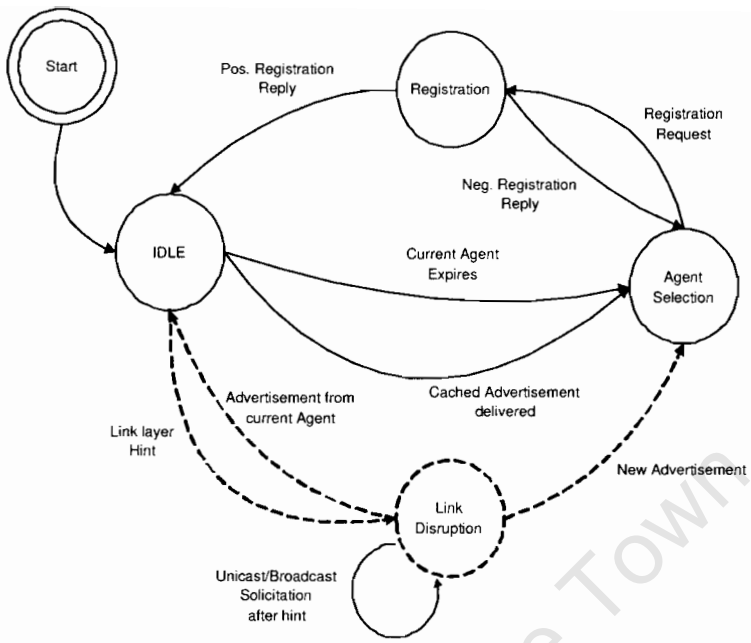

Figure 4.9: Advertisement-caching finite state machine

One of the advantages of using a caching agent is that no modifications have to be made to a MN. A MN can use the default Mobile IP eager-binding policy and a cached unsolicited advertisement is delivered as soon as it becomes associated with an AP.

4.6 Design Comparison

The following list compares the most important factors associated with the movement detection optimisation designs described above.

- Movement detection should be performed as quickly as possible. However, optimisations should perform reliably, especially considering the variability of wireless connections and handoffs.
- Link-up hints should be triggered by the reception of a positive 802.11 (re)association response or a change in AP identity (MAC address). A change in ESSID *may* be used, but this technique is somewhat less reliable.

- Broadcast solicitation/advertisement exchanges should be avoided where possible. Even when advertisements are replied to immediately, this exchange introduces delays. Furthermore, these exchanges should be particularly avoided in 802.11 wireless LANs.

- Modifications of 802.11 elements that allow additional IP information to be transported are extremely difficult to carry out. Proposals that make use of these modifications cannot be implemented using current commercial devices.

Both the advertisement caching scheme and FHCS are expected to result in the fastest movement detection because new advertisement information is immediately delivered to MNs. This is an important factor because VoIP is very sensitive to the bursty packet loss caused by loss of connectivity. The FHCS technique provides advertisement information instantaneously (without any additional signalling), but is difficult to implement when relying on newly defined 802.11 IEs.

On the other hand, the caching agent solution incorporates all the factors listed above. Furthermore, this technique supports MNs that do not have link layer hint facilities. Reliability is improved because caching agents support two modes of operation: advertisement caching and HCS. The HCS functionality provides MNs with a fall-back mechanism should the cached advertisement be missed. When all these factors are considered, the advertisement caching scheme, using a separate caching agent, provides the most efficient and complete solution.

4.7 Hybrid Technique

4.7.1 Link Layer Hints

The hybrid technique incorporates several different mechanisms to ensure fast and robust movement detection. Based on the conclusions of the previous section, the hybrid technique uses a caching agent to deliver advertisements to MNs. 802.11 hints are therefore used by both the network and the MNs. On the access network, the caching agent uses link-up hints from 802.11 APs to replay cached advertisements. On the MN, both predictive hints and event hints are used to select the agent selection policy (eager/lazy-binding). Figure 4.10 illustrates the different hints that are used within the hybrid system.

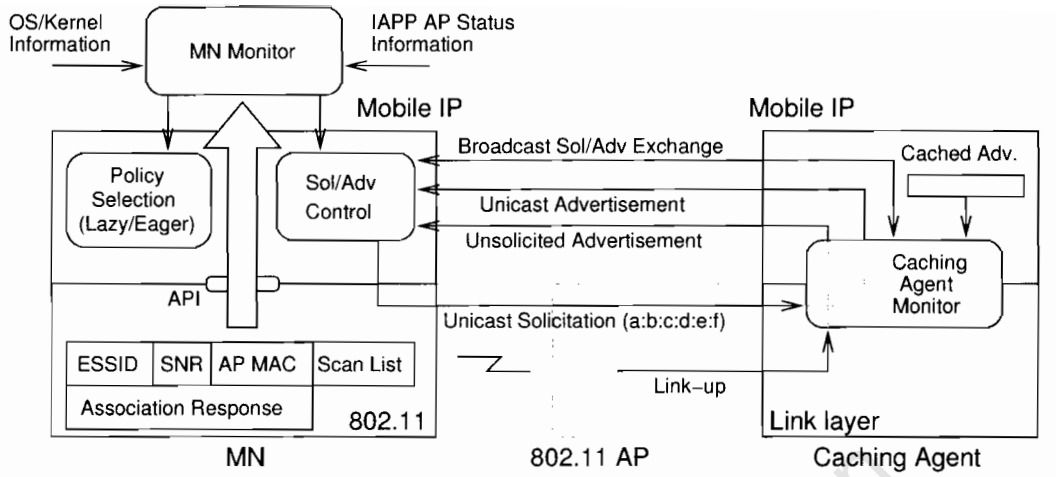

Figure 4.10: The hybrid movement detection system

This mechanism relies on the assumption that terminal movement gradually changes the wireless link quality and that an 802.11 handoff is initiated based on these parameters [77, 3]. As described in the previous chapter, the MN periodically monitors the quality of the link to the current AP. When the quality falls below a certain threshold, the eager-binding agent selection policy is activated in anticipation of a link layer handoff.

The MN uses a separate hint monitor to generate link layer hints. The monitor also translates these hints into appropriate Mobile IP functionality (e.g. a change in agent selection policy). This MN monitor has access to scan results from the wireless card's driver, allowing a list of neighbouring APs to be compiled. For example, under the Linux OS, an 802.11 scan could produce the following results:

```
eth2 Scan completed :
Cell 01 - Address: 00:0D:88:17:0A:9D
    ESSID:"crg"
    Frequency:2.442GHz
    Quality:0/10 Signal level:-29 dBm Noise level:-256 dBm
Cell 02 - Address: 00:0D:88:50:70:17
    ESSID:"crg"
    Frequency:2.442GHz
    Quality:0/10  Signal level:-42 dBm  Noise level:-256 dBm
Cell 03 - Address: 00:02:2D:01:3C:BE
    ESSID:"home"
    Frequency:2.447GHz
    Quality:0/10 Signal level:-49 dBm Noise level:-256 dBm
```

The MN can therefore access a list of MAC addresses for all scanned APs, along with additional information from the link layer (such as ESSID and frequency). These scan results are accessed periodically by the MN monitor and will allow a list of neighbouring AP information to be produced. An example of such a list (created from the scan results above) is shown in Table 4.2:

Subnet Prefix	ESSID	AP MAC Address	Signal level	Status
prefix1	crg	00:0D:88:17:0A:9D	-29 dBm	on-link
prefix1	crg	00:0D:88:50:70:17	-42 dBm	on-link
prefix2	home	00:02:2D:01:3C:BE	-49 dBm	-

Table 4.2: Neighbouring 802.11 AP information

The MN monitor also detects event hints from the operating system or wireless card driver using the mechanisms discussed in previous sections. After a MN performs an 802.11 handoff, the MN monitor receives a link-up hint that includes the new AP's MAC address. If the AP resides on the same IP network as the previous AP, the lazy-binding policy is activated. If the MN verifies that the new AP is not on the same link or is unsure, the eager-binding policy is used.

4.7.2 802.11 Hint API

The 802.11 hint monitors described above should ideally receive link layer information via an application programing interface (API). This allows link layer parameters to be queried in a systematic and portable fashion. For example, a Linux OS kernel supporting "wireless extensions" allows applications to access an array of wireless parameters and statistics. In order to support reliable 802.11 handoff hints, both wireless device drivers and operating system APIs must be extended to provide more link layer process information. For example, the current stage of an 802.11 handoff (scan, authentication etc.) or events such as the reception of a positive association response message should be made available to the monitors. Unfortunately, implementing these extensions is a very lengthy process and requires the cooperation of device manufacturers. As a result, this study will not develop these extensions and will rely on nonstandard mechanisms to generate hints.

4.7.3 Functional Design

All of the information in Table 4.2 is retrieved from the wireless API, with the exception of the status and subnet prefix fields. The status and prefix fields indicate whether a given AP is on the current IP network. One possible way that this can be achieved is by extending the IAPP protocol to allow a MN to obtain this information. A MN could therefore query an IAPP Radius server and either request a list of on-link APs or verify individual AP MAC addresses. This would typically be performed concurrently with other upper-layer traffic, during periods where the MN's wireless connection has spare capacity.

The hybrid mechanism includes a certain degree of redundancy. Before an 802.11 handoff is performed, the eager-binding is activated when the link quality passes below a specific threshold. After link layer handoff is complete, the identity of the new AP is verified and the eager-binding is deactivated if the MN is still attached to the same network. It may seem unnecessary to activate eager-binding before a handoff and then subsequently deactivate it after handoff. The reason for specifying this behaviour is to improve reliability.

The wireless medium is highly unpredictable and a MN may handoff to an AP that has not been previously detected. Therefore, even if an intra-subnet link layer handoff seems imminent, an eager-binding should still be used. Consider that a layer 2 handoff can be performed within a subnet using an eager-binding without introducing any delays. However, if a lazy-binding is mistakenly activated for an inter-subnet movement, the resulting delays will negatively affect VoIP sessions. This is why, as a precaution, the MN should always activate an eager-binding before an 802.11 begins and a lazy-binding should be activated after the handoff, when the location of the AP is verified.

Furthermore, a sudden change in wireless conditions may induce a handoff before it can be predicted by the MN monitor. The hybrid system will still function correctly even under these conditions. When the link-up hint is received, a lazy/eager-binding policy is activated depending on the AP's MAC address. Layer 2 handoff prediction merely provides an additional level of reliability.

Chapter 5

Evaluation Framework Architecture

5.1 Introduction

This chapter describes the underlying hardware and software platform used to support the evaluation of different movement detection techniques (as discussed in previous chapters). This evaluation framework allows both Mobile IP handoff and movement detection performance to be assessed using commercial 802.11 equipment. It also supports VoIP systems, and illustrates the effects that different movement detection techniques have on the resulting output voice quality.

Several abstractions have been made in the development of the evaluation framework. For example, commercial Mobile IP mobility agents are not used within the framework due to the fact that their functionality cannot be modified or extended easily. Mobility agents are therefore emulated by applications operating on generic desktop computers. The implications of such abstractions will be discussed as they are introduced.

At the outset, a general description of the evaluation framework will be given. The mobility management software used within the framework will be introduced and the reasons for selecting this particular Mobile IPv4 implementation will be provided. A more detailed description of the different entities within the framework will follow, together with an outline of the evaluation experiments that have been performed. The evaluation framework supports two physical configurations. The first is used to test movement detection techniques other than the hybrid system. The hybrid system incorporates aspects from many other optimisations and a second network configuration is used to highlight these factors. As a result, special attention is given below to the implementation of the hybrid system.

5.2 Evaluation Framework Overview

Figure 5.1 illustrates the wireless access network model that has been progressively developed throughout the previous chapters. The model includes a hierarchical micromobility architecture that ensures all registration signalling is isolated within the domain. Inter-domain terminal movement will not be considered due to the additional delays that are introduced when registration messages are forced to travel outside of the local domain. A MN uses an 802.11 wireless network card to connect to a particular IP subnet (i.e. either a home or foreign network). The MN is also able to establish a VoIP connection to a corresponding node using the access network.

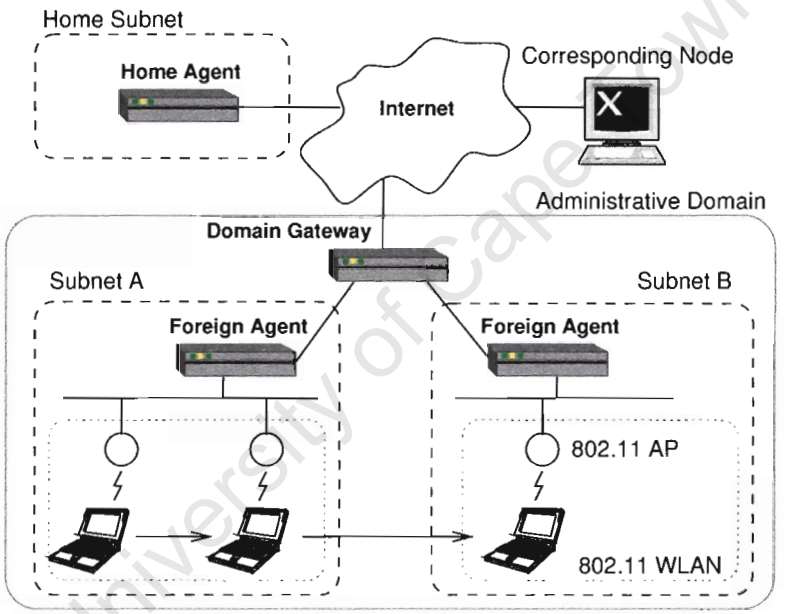

Figure 5.1: Wireless access network architecture

When this model is translated into a practical testing framework, a number of useful simplifications can be incorporated. Firstly, the evaluation framework is a small-scale network with relatively little complexity. This is an important design criterion as the main focus of this project is isolated to specific Mobile IP handoff processes. It is therefore beneficial to limit the influences of other factors (such as increasing numbers of MNs) because this results in simpler data analysis. Therefore, the home and foreign IP networks are not connected through the Internet, as is the case in Figure 5.1. The different entities that make

up the framework are also not separated by large physical distances and do not support high traffic loads. As a result, Mobile IP registration delays within the evaluation framework are relatively small and the hierarchical micromobility protocol/architecture is not necessary. Another simplification results from the integration of 802.11 AP functionality into the foreign agents, which will be discussed in further detail below. Figure 5.2 depicts the structure of the evaluation network that will be used to emulate intra-domain Mobile IP handoffs.

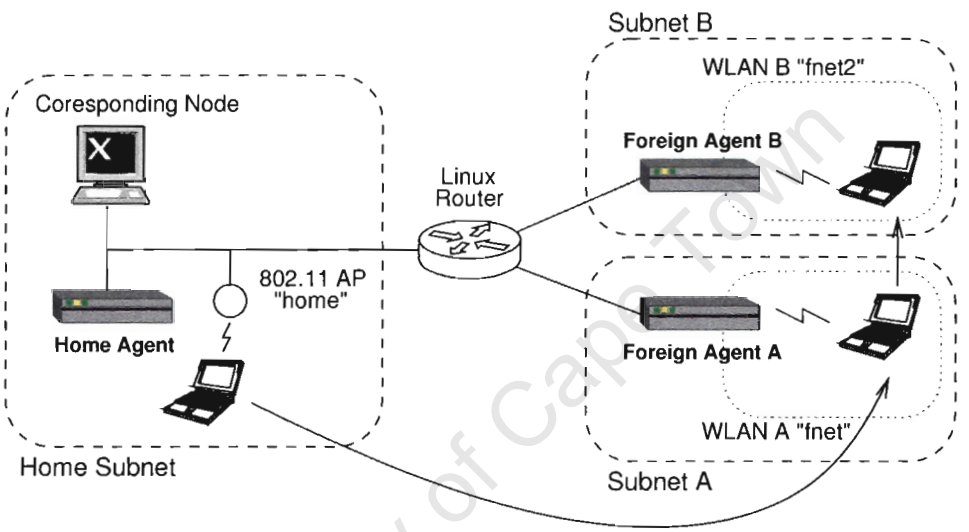

Figure 5.2: Evaluation framework architecture

The evaluation framework consists of one home and two foreign IP networks. Instead of using the Internet to transport IP packets between the home and foreign IP subnets, these networks are interconnected via a simple software router. Each foreign network contains a single foreign agent. A desktop computer is used to implement the MN and terminal mobility is emulated by forcing its wireless card to associate with different APs. Because the APs are all on different IP networks, this initiates a Mobile IP handoff. For simplicity, the MN only communicates with a corresponding node located on its home network. Table 5.1 below describes the hardware used to implement each entity within the framework. It will be shown that limitations in the computing resources of certain entities have a significant effect on Mobile IP handoff latency.

Framework Entity	Hardware System
Home agent	Pentium I, 200 MHz, 32 MB RAM
Linux router	Pentium I, 166 MHz, 64 MB RAM
Foreign agents	Pentium II, 400 MHz, 64 MB RAM Agere Systems Orinoco 802.11b wireless PCMCIA card
Mobile node	Pentium III, 730 MHz, 256 MB RAM Agere Systems Orinoco 802.11b wireless PCMCIA card
Corresponding node	Pentium IV, 1.4 GHz, 512 MB RAM

Table 5.1: Evaluation framework hardware platform

5.3 Dynamics Mobile IPv4

5.3.1 Overview

Most movement detection optimisations extend existing Mobile IP mechanisms because they aim to remain compatible with the Mobile IP specifications. It is therefore unnecessary to design and build an entirely new Mobile IP implementation. Several universities and institutions have developed implementations of the Mobile IPv4 specification to varying degrees. A number of these implementations have been carefully studied, although a full comparison of their features is beyond the scope of this document. The Helsinki University of Technology (HUT) has developed a full-featured Mobile IP implementation called "Dynamics" [6] which is used to support Mobile IP mobility management within the evaluation framework. This particular implementation was selected for the following reasons:

- Dynamics is an open-source project which allows the source code to be extended.
- The Dynamics software package was designed for the Linux operating system. The advantages of using the Linux OS are that interactions between Dynamics processes and the operating system are very flexible.
- All Dynamics processes are executed in *user-space* as opposed to *kernel-space*. This allows the source code to be more easily modified. The reason for this is that program code that is executed in the kernel is more difficult to extend and debug[1].

[1]User-space code may use pre-compiled libraries whereas kernel code may only use special system calls.

- One of the most important reasons for selecting Dynamics is that it includes various libraries that provide well-defined APIs to external application processes. A user application may access mobility information and initiate Mobile IP functions through these libraries. The APIs allow different extensions to the default movement detection processes to be implemented.

- Dynamics can be used to implement a micromobility architecture as it supports the establishment of a hierarchy of foreign agents. Although this feature is not used in the current framework, it may be useful if the evaluation framework is expanded into a much larger architecture in the future.

- Lastly, in contrast to several other implementations the Dynamics software is easy to compile and install. It is also relatively well documented and the project web page [6] supports a helpful mailing list.

5.3.2 Dynamics API

Three important mobility management programs can be compiled using the Dynamics source code, along with several utilities, sample code and testing tools. The programs are executed on a machine as background processes or *daemons*. Each daemon (dynhad, dynfad and dynmnd) implements home agent, foreign agent or MN functionality. For example, the MN dynmnd daemon listens for agent advertisements and performs registration signalling if necessary. Only one of these daemons is typically executed on a given machine based on its desired function (MN, home or foreign agent). The Dynamics API provides several libraries of C functions that allow external processes to communicate with these daemons. External processes are thereby able to monitor mobility information and modify dynamic parameters in order to alter the behaviour of these daemons. The following list provides a brief overview of some of these functions [12]:

Foreign and Home Agent API (dynfad/dynhad)

- List active tunnels

- Read a specific tunnel's information

- Destroy a tunnel

Recall that a home agent forwards MN traffic to a foreign agent through a tunnel by encapsulating packets. These APIs allow an external process to query the mobility agent daemon and determine which tunnels are actively carrying traffic. Specific tunnel information will include the source and destination end-points of the tunnel along with the tunnel's remaining lifetime. Lastly, a tunnel can be destroyed, whereby the mobility agent no longer forwards traffic for the associated MN.

Mobile Node API (dynmnd)

- Get current CoA
- Get current agent status
- Connect/disconnect from a specific agent
- Update location (force a new agent IP address to be used)
- Change selection policy (ECS/LCS)

These API functions allow a user application running on a MN to determine the identity of a visited network along with the MN's current mobility status. For example, the application can determine whether the MN is receiving traffic or attempting to register with a new mobility agent. The API also allows specific mobility agents or IP networks to be selected via the *update location* procedure. For further information on the Dynamics APIs, refer to the "Dynamics Functional Description" [12] or the API source code itself.

5.4 Framework Entities

5.4.1 Foreign Agents

The foreign agent entities integrate both 802.11 AP and Mobile IP foreign agent functionality. The system overview of a foreign agent is given in Figure 5.3.

Foreign agents are each equipped with a wired and a wireless network interface. The wired interface is provided by an Ethernet link to the rest of the foreign network. A commercial 802.11 PCMCIA wireless card establishes the second interface and performs

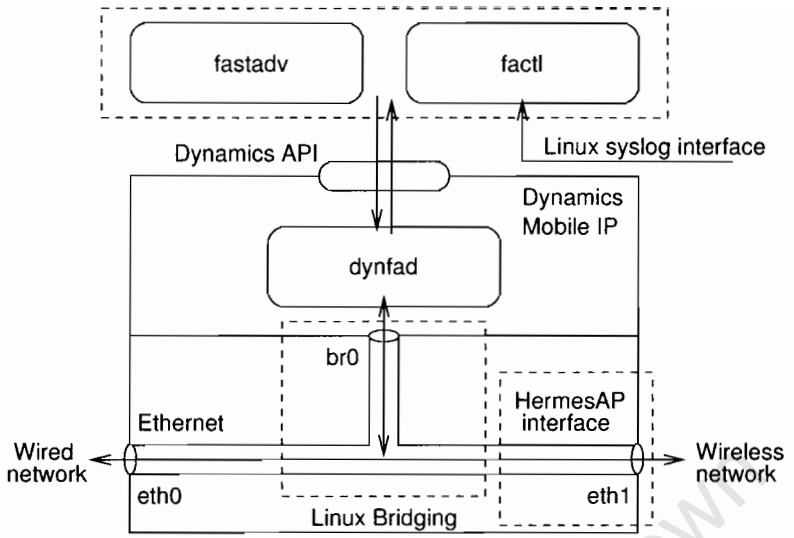

Figure 5.3: Foreign agent system overview

access point functionality. Ordinarily, wireless cards do not have native support for such functionality within their firmware. Instead, they are designed to have simple station capabilities. HermesAP [1] is a Linux utility that allows an 802.11 wireless card (based on the "Hermes" chip-set) to operate in access point mode. This allows the wireless card to perform AP functionality such as allowing associations from neighbouring stations and beacon broadcasting. The HermesAP *hwload* tool is used to download third-party AP firmware to the RAM of the wireless card. Once this download is complete, the card executes the AP firmware and emulates the AP's behaviour. The HermesAP package also includes a modified device driver that supports this AP firmware.

HermesAP only allows an 802.11 interface to support AP station functionality. However, a conventional 802.11 AP also acts as a bridge between the (wired) distribution system and the wireless medium. A foreign agent entity performs this bridging using Linux link layer bridging tools (*ebtables* and *brctl*) [7, 2]. When link layer bridging support is patched into a Linux kernel, these tools allow two or more physical network interfaces to be bridged (see Figure 5.3). The bridge is assigned its own logical interface (br0) through which the Dynamics foreign agent process can access both interfaces. When a frame enters any interface (physical or logical) of a link layer bridge, it is automatically forwarded to all other interfaces.

The main reason for combining foreign agent and 802.11 AP functionality is that this facilitates simple hint communication between 802.11 and Mobile IP entities. However, another motivation is that the available commercial access points do not forward broadcast traffic at all. It is assumed that this behaviour is an extension of the IEEE 802.11 recommendation for broadcast traffic (refer to section 3.6.3). Further investigation is required to determine whether this behaviour has been implemented by all commercial access points[2].

In a Dynamics foreign agent, the dynfad daemon is usually responsible for broadcasting periodic advertisements, responding to solicitations and forwarding registration messages. In addition, within the evaluation framework, the MN uses a foreign agent CoA provided by the dynfad process when visiting a foreign network. Therefore, the dynfad daemon also acts as the endpoint of Mobile IP tunnels and delivers decapsulated packets to the MN using the link layer. The justification for using the FA CoA addressing mode has been discussed in previous chapters (refer to section 3.4). For most movement detection techniques, the dynfad process performs all of these functions.

However, in order to implement the advertisement caching technique, caching agent functionality has been included within the foreign agent entities. This is achieved by using an external *factl* monitoring and control process. In this configuration, the dynfad daemon performs all the functions described above, except for broadcasting advertisements. The factl process is responsible for sending periodic advertisements using the Dynamics FA API. Instead of caching and replaying advertisements, the factl emulates caching agent functionality by immediately transmitting an advertisement when a link layer hint arrives.

The factl process receives link layer handoff hints directly from the wireless device drivers through the *syslog interface*. The HermesAP wireless device drivers log 802.11 handoff events (such as disconnection and successful association). These events are passed to the Linux system logging facility (syslog) where they are recorded in appropriate system log files. However, the syslog subsystem also passes this log information to a FIFO pipe, thereby allowing running processes to receive notifications of these layer 2 events. A FIFO pipe is essentially a special file[3] that allows information to be communicated in real-time between separate processes (i.e. the syslog subsystem and the factl process) [54]. When the factl process detects a successful association event using the syslog interface, an advertisement is broadcast to the newly associated MN. Figure 5.4 illustrates the main aspects of the factl syslog interface.

[2]APs from only one manufacturer were available for testing.

[3]In this case, the `/dev/xconsole` FIFO was used. Users may also define their own FIFO files.

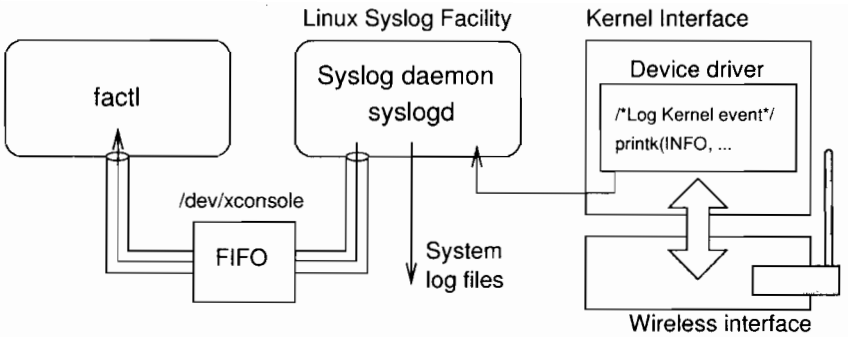

Figure 5.4: Factl syslog interface

As Dynamics conforms to the Mobile IPv4 specification, the foreign agent daemon inserts random delays into every solicitation/advertisement exchange. In order to implement and test the FastADV technique (in conjunction with HCS), foreign agent entities execute a *fastadv* process. The fastadv process uses the foreign agent API to detect solicitations and reply to them immediately. Fastadv is also executed independently from the factl process and the two can be run concurrently. These additional systems are depicted in Figure 5.3 within the upper dashed rectangle.

5.4.2 Linux Software Router

The software router includes three network interfaces, each connected to either a home or foreign network. Again, the Linux operating system was chosen to implement the router because of its well-integrated IP packet filtering tools. The *iptables* tool was used to establish simple rules for filtering and forwarding IP packets. This ensures that each IP subnet is both physically and logically segregated from the other networks, as they would be within the Internet.

The Linux software router has been kept as rudimentary as possible. The capacity of such a simple router is extremely limited (especially considering its minimal system resources). The software router also only supports static routes, specified manually. With only three IP networks under consideration, it is unnecessary to include more sophisticated routing functionality. Despite the severe limitations of this network entity, it adequately supports the evaluation framework's simple network configuration.

5.4.3 Home Agent

On the home agent, the dynhad daemon performs all home agent functionality such as maintaining a binding cache, tunnelling packets to its MNs and handling registration exchanges. An additional "home agent controller" (*hacntl*) process is also executed on the home agent. The purpose of the controller is simply to ensure that the dynhad daemon is correctly configured when the MN performs handoffs. Handoff experiments are repeated a number of times to ensure that meaningful results can be derived. The home agent controller therefore reinitialises the dynhad configuration parameters (such as tunnel state) before every experiment.

The home agent also suffers from the same capacity limitations as the Linux router. This is mainly due to the fact that the underlying hardware platform includes limited processing and memory resources. The capacity of a home agent is usually of extreme importance because all the MN's traffic must flow through its home agent while it is away from home. The home agent therefore has the potential of becoming a bottleneck in a Mobile IP system[4]. Despite these issues, the home agent is able to effectively support a single MN with relatively low traffic levels.

5.4.4 Mobile Node

Much like the foreign agents, the wireless card driver on the MN also reports successful association events to the syslog subsystem. Thus, the MN also uses a syslog interface to determine when an 802.11 handoff is completed. This allows link layer handoff latency to be evaluated quantitatively. In addition, the syslog interface is able to initiate Mobile IP functions, such as solicitation/advertisement exchanges, immediately after an 802.11 handoff via the Dynamics MN API. This is used to support the Hinted Cell Switching (HCS) and FastADV optimisations. The MN syslog interface is similar to the FA version described above (Figure 5.4).

Unfortunately, the Fast Hinted Cell Switching (FHCS) movement detection optimisation could not be implemented on the evaluation framework. As described in the previous chapter, this technique is extremely difficult to implement using new application-specific information elements. In addition, FHCS could not be implemented using the simpler method of transporting a foreign agent's IP address within the ESSID of a wireless LAN.

[4]A technique for alleviating this problem was discussed briefly in Chapter 2.

The failure of this mechanism can be attributed to the limitations of the Dynamics MN API. The MN API does not allow external programs to directly set foreign agent information without verification. What this means is that an external process may force the MN to register with a particular FA by submitting the FA's IP address through the *dynamics_mn_force_fa*() or *dynamics_mn_update_location*() API calls. However, the API always verifies these IP addresses by performing a solicitation/advertisement exchange with the candidate FA. This extra exchange is precisely the step that the FHCS technique aims to avoid. The Dynamics API source code could not be successfully modified to support FHCS functionality.

The MN dynmnd daemon source code has been modified to allow the generation and capture of detailed timing information as the MN changes states. Timestamps of all significant events are saved and used to perform an "off-line" analysis of the different constituent delays in a Mobile IP handoff.

5.4.5 Timer Resolution

All the elements in the evaluation framework have been implemented using the Linux operating system. One of the limitations of this OS is that its maximum timer resolution is 10 ms[5]. For example, the maximum rate that a periodic timing interrupt can be triggered is once every 10 ms. However after consulting several Linux-related mailing lists and other online resources, it was discovered that certain functions, such as *gettimeofday*(), offer much higher time granularity [30]. The reason for this difference is that these functions return the absolute calendar time using the (on-board) system clock, unlike system timers that return a time interval. It has been reported that these time functions support millisecond timing resolution, which is adequate for performing movement detection and handoff evaluations.

5.5 Hybrid System Configuration

The hybrid system relies on the coordination of several movement detection optimisations. Therefore, most of the details regarding the hardware and software systems that support this technique have been covered in the previous sections. The MN monitor is responsible for controlling and coordinating these different subsystems. The MN monitor is a user-level

[5]These intervals are termed "jiffies".

process that accesses both 802.11 and Mobile IP information through their respective APIs. The MN monitor is also able to use the Dynamics API to initiate certain MN functions, such as agent policy selection and solicitation/advertisement exchanges. An overview of the hybrid system is depicted in Figure 5.5. The implemented hybrid system does not include all of the functionality discussed in the previous chapter. Excluded functionality has been marked in grey.

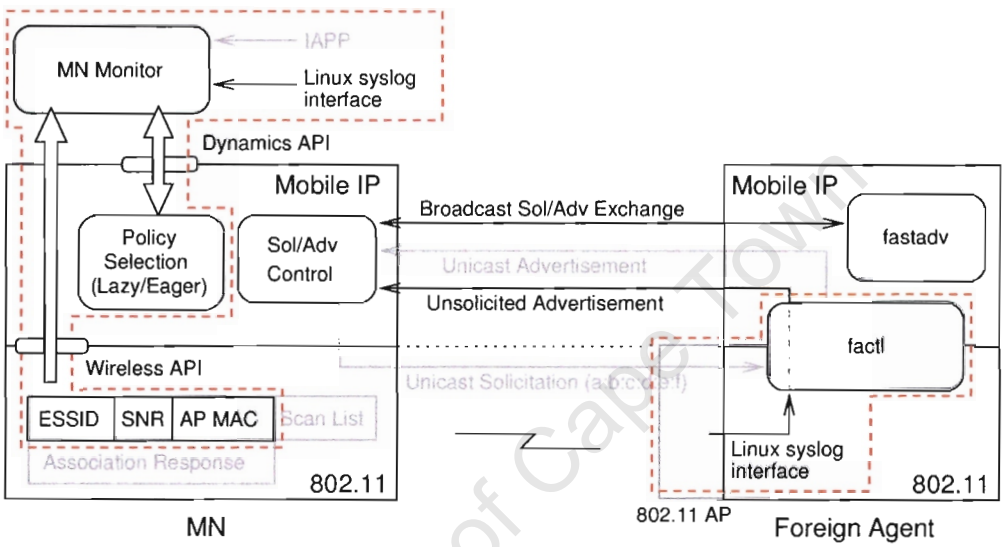

Figure 5.5: The hybrid movement detection system

The MN uses its syslog interface to determine when 802.11 handoffs occur. When these handoffs are detected, the MN picks the most appropriate selection policy. The MN monitor determines which policy to activate based on information from the wireless API. The monitor uses the new AP's MAC address to ascertain whether it is on the same network as the previous AP. It was proposed in Chapter 4 that this could be achieved using 802.11 scan information along with an extended IAPP protocol. However, due to project time constraints this mechanism has not been implemented and the mapping from AP MAC addresses to IP subnet has been hard-coded into the monitor source code.

The MN monitor also continuously probes 802.11 link quality statistics in an attempt to anticipate link layer handoff. The Orinoco wireless cards used in the evaluation framework define their cell search threshold in terms of the wireless link's signal-to-noise ratio

(SNR). When the link SNR drops below 10 dB, an intra-ESS[6] handoff is initiated [77, 4]. Therefore, in addition to the syslog interface, the MN monitor uses the SNR to determine the appropriate selection policy. When the link SNR is decreasing and approaching the cell search threshold (including suitable hysteresis), the eager-binding policy is activated. Likewise, when the SNR climbs sufficiently above this threshold, a lazy-binding is used. This explanation should clarify the fact that hint generation in the hybrid system employs a combination of the monitoring and device driver models. For example, the MN monitor periodically queries information from the link layer in addition to receiving device driver events via the syslog interface. The hybrid system also incorporates both predictive (although they are somewhat unreliable) and event hints.

On the foreign agent entities, the factl process emulates the caching agent functionality by broadcasting an agent advertisement as soon as a MN successfully associates with its wireless interface. The fastadv process replies immediately to any received solicitation messages. As described previously, these functions use a syslog interface and the Dynamics API. However, these details have been omitted from the foreign agent diagram in Figure 5.5 for the sake of clarity. Figure 5.5 also illustrates that the unicast solicitation/advertisement exchange is not supported by either the MN or FA. The reason for omitting this functionality is that, although it is an enhancement to the overall system, this exchange does not explicitly improve movement detection performance.

Because the hybrid system builds onto previous designs, only certain key aspects need to be tested. These core subsystems are enclosed by broken lines in Figure 5.5. The broadcast solicitation/advertisement functionality and the fastadv process are not part of the essential hybrid system, but rather are useful extensions. These peripheral subsystems are also evaluated as part of other movement detection schemes and will not be considered in this section. Furthermore, in order to test the hybrid system effectively, the evaluation framework must assume a slightly different configuration. These changes ensure that the most important attributes of this mechanism are highlighted. The modified evaluation framework architecture is shown in Figure 5.6.

As with the previous configuration, this architecture allows a MN to perform handoffs between different wireless LANs residing on separate IP subnets (movement 2 and 3 in Figure 5.6). These handoffs are usually triggered by the user when an ESSID is changed. However, the modified evaluation framework also includes an IP subnet with two FA/AP entities. This allows the MN to "move" between different foreign agents and 802.11 APs

[6]Recall that an intra-ESS handoff occurs between APs using the same ESSID.

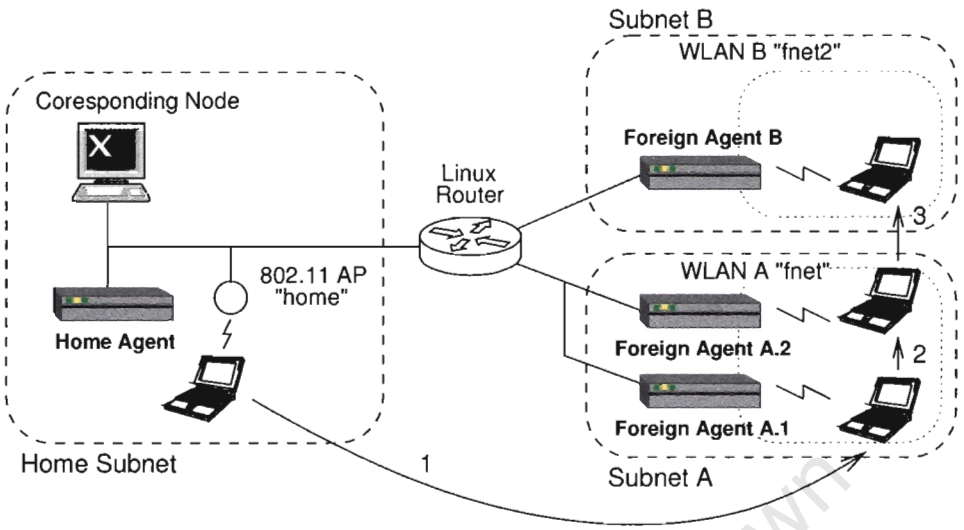

Figure 5.6: Modified evaluation framework architecture

on the same network (movement 2). In this case, 802.11 handoffs are triggered by changes in the link quality. The hybrid system should therefore switch appropriately between the eager and lazy agent selection policies, irrespective of what type of link layer handoff has occurred (inter or intra-subnet).

5.6 Evaluation Experiments

The different movement detection techniques investigated in this study are tested on the evaluation framework in two phases. In the first phase, movement detection mechanisms are evaluated without any traffic load. These evaluations are more general, in that the contributions of different stages of a Mobile IP handoff to the total handoff latency are determined. These results include 802.11 handoff, movement detection and registration latency. In the second phase, a VoIP application is used to ascertain how different movement detection mechanisms affect the output voice quality. Important factors such as the packet loss caused by handoffs are considered. This phase will illustrate the extent to which different movement detection schemes can support VoIP systems.

5.6.1 Handoff Tests (Phase 1)

The following list describes the handoff tests used to evaluate Mobile IP handoff in general, along with different movement detection optimisations:

- The generic advertisement-based Mobile IPv4 movement detection schemes are tested. The default lazy and eager-binding mechanisms provide a reference for comparing the improvements that result from movement detection optimisations.
- The HCS mechanism is evaluated initially with FastADV functionality disabled. FastADV is then enabled and the HCS tests are repeated to measure any improvements contributed by the FastADV technique.
- The emulation of the advertisement caching optimisation is assessed.
- For each movement detection scheme, the MN performs layer 2/3 handoffs by forcing its wireless card to associate with different APs. This is achieved by changing the card's ESSID. The following movement pattern is repeated 10 times: home network, foreign network A, foreign network B. This entire procedure is then repeated using several different advertisement rates. The end results will verify whether movement detection optimisations are independent of advertisement rate in practice.
- Lastly, the hybrid system is tested within the modified evaluation framework, allowing its theoretical performance and stability improvements to be verified.

5.6.2 VoIP Tests (Phase 2)

The second phase essentially evaluates the packet loss and degradation of output voice quality induced by different movement detection schemes.

- A VoIP connection can be modelled by a stream of packets with a well defined packet-size and transmission rate. A simple client/server application has been developed to ascertain the packet loss that results from a Mobile IP handoff. The server (*dgram-gen*) streams numbered UDP packets to the client (*dgram-rec*). A typical PCM voice codec is emulated by padding each packet with 172 Bytes[7] of data and transmitting

[7]For the PCM G.711 codec: 160 Byte payload + 12 Byte RTP header

a packet every 20 ms [76]. The client calculates the interval between consecutive packets and records any delayed or lost packets.

- The deterioration of output voice quality due to a Mobile IP handoff can be evaluated using the Robust Audio Tool (RAT) [63]. RAT is a flexible VoIP application that allows point-to-point voice calls to be established. RAT supports many useful features, including a variety of different voice codecs. In these experiments, the RAT transmitter reads voice data from a sound file and sends it over the evaluation framework. The RAT receiver saves the incoming voice stream to an output file. The input and output files can then be compared to determine the qualitative deterioration.

- Experiments only consider data flowing in one direction at a time. This is based on the assumption that (usually) only one party in a VoIP call is speaking at a time (termed "talkspurt") and that a handoff will only affect one party's traffic [53].

- During preliminary tests it was discovered that Mobile IP handoffs within the evaluation framework affect up-link and down-link traffic in very similar ways. For this reason, experiments are performed using only down-link traffic.

- Lastly, these results will be correlated with the results from the first phase to ensure that they are consistent.

Chapter 6

Evaluation Results and Analysis

6.1 Introduction

This chapter evaluates the performance of different movement detection optimisations within the evaluation framework. Results of different handoff experiments are presented and will be used to compare the improvements that these optimisations contribute to Mobile IP handoff. Initially, general Mobile IP handoff characteristics, including registration and link layer handoff delays, will be investigated. The performance of the default Mobile IP movement detection mechanisms will also be assessed. The main motivation for developing movement detection optimisations is to reduce the delays introduced by existing mechanisms. However, a secondary reason is to avoid broadcasting advertisements at high rates. Therefore, the effects of high broadcast rates within 802.11 wireless LANs will be discussed briefly.

In addition to measuring the numerical timing performance of different movement detection techniques, the evaluation framework also allows VoIP disturbances at the application level to be assessed. The framework has been designed to facilitate simple translations of these user-perceived artifacts to their root causes at the network level. In order to achieve this, the output quality of the VoIP system can be assessed either subjectively by using human judgement or objectively through analytical methods.

The rigorous assessment of a VoIP audio stream that has been subjected to packet loss is a complicated and intricate matter that lies beyond the scope of this study[1]. Speech

[1]This is in contrast to the well-defined end-to-end delay and jitter requirements.

impairments are affected by factors such as the audio codec used, loss pattern and position of the loss [73]. Therefore, for the purposes of this study, the output voice quality during a Mobile IP handoff will be evaluated subjectively using user-perception techniques. Although these techniques are not very comprehensive, qualitative results can be used to provide a crude indication of how effectively Mobile IP/802.11 access networks support VoIP traffic.

The experiments performed on the evaluation framework have been divided into three sections. The first section deals with the evaluation of Mobile IP handoff and the different movement detection techniques. The primary focus of this phase is to investigate the latency of these processes. In the second phase, an evaluation of the VoIP system is performed using the output speech quality and packet loss as performance metrics. Lastly, the hybrid system is assessed within the modified evaluation framework.

6.2 802.11 Handoff

In order to isolate and evaluate 802.11 handoff latency, 100 link layer handoffs were performed between two overlapping APs. Each inter-ESS handoff was initiated by changing the wireless card's ESSID. As a result, the card began scanning for a new AP using the specified ESSID and followed the procedures outlined in Chapter 2 to establish a new link. These tests did not investigate the different components of an 802.11 handoff, but rather considered the link layer handoff process as a whole. The following results were obtained:

Average handoff latency	156 ms
Standard deviation	30 ms

It is important to note that these results are highly dependent on the equipment used. The handoff properties listed above were produced using a single hardware configuration consisting of an Agere Systems Orinoco 802.11b wireless card and HermesAP access points. Both HermesAP wireless cards use exactly the same firmware revision which ensures that their behaviour (especially during handoff) is as consistent as possible. Furthermore, an Open System 802.11 authentication mechanism has been used throughout these experiments. A Shared Key mechanism, on the other hand, would involve additional message exchanges that may increase the total 802.11 handoff latency. Nevertheless, the above results fall well within the wide range of values reported by other studies (refer to section 2.4.4).

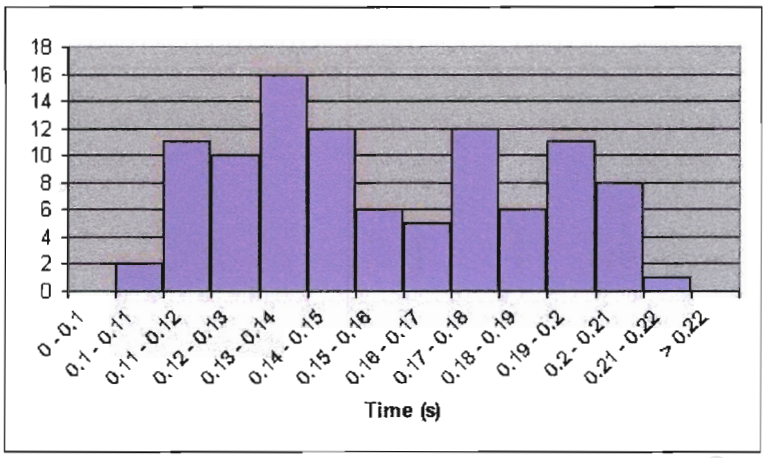

Figure 6.1: Distribution of 802.11 handoff latencies

An interesting feature of these results is the significant variance in individual 802.11 handoff latencies, as indicated by the relatively large standard deviation. Figure 6.1 illustrates the relative distributions of these handoff latencies over the 100 tests. From the figure, 802.11 handoff latencies are mainly distributed within two intervals: between 110 - 150 ms and 170 - 210 ms. The factors that produce such a wide range of delays are difficult to assess as all link layer processes are hidden.

6.3 Advertisement – Bandwidth Trade-off

A theoretical estimate of the bandwidth consumed by periodic Mobile IPv6 advertisements was presented in Chapter 3. A quantitative measurement of Mobile IPv4 advertisement bandwidth usage was then performed within the evaluation framework. Agent advertisement frame sizes at the link layer were measured using the Ethereal packet analysis tool [8]. The corresponding bandwidth utilisation at different advertisement intervals is shown in Figure 6.2.

A Mobile IPv4 foreign agent advertisement that contains only a FA CoA (without any further extensions) occupies 110 Bytes at the Ethernet/802.11 link layer (96 Byte IP packet). The bandwidth consumed by these advertisements at the Mobile IPv6 50 ms minimum interval is approximately 17.6 kbps. This is reduced to under 200 bps when an interval of 5 seconds is used. When considering the fact that 802.11b wireless LANs theoretically

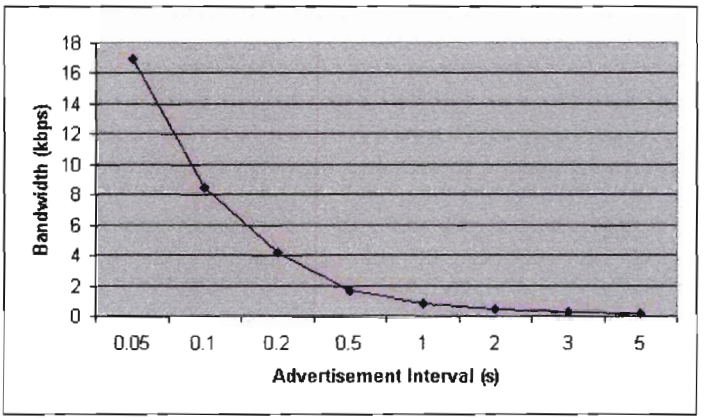

Figure 6.2: Advertisement bandwidth usage

offer 11 Mbps of bandwidth, the resources wasted by advertisements are negligible, even at the highest broadcast rate.

In practice however, a number of secondary issues make high advertisement broadcast rates undesirable. Firstly, as the number of MNs per access point increases, the observed throughput decreases dramatically (non-linearly). This is due to the 802.11 DCF mechanism where the wireless medium is reserved by individual stations before they are allowed to transmit [59]. As a result, stations are forced to endure higher medium access delays when the wireless medium is shared in this way.

802.11 dynamic rate scaling decreases the bandwidth available to a MN even further when the physical distance or radio attenuation between the AP and MN increases. Consider that 6 MNs attached to the same AP through 1 Mbps wireless links are reported to achieve a maximum data throughput of 16.7 KBps each [59]. Large amounts of broadcast advertisement traffic will further restrict this maximum throughput. The 802.11 DCF also introduces significant medium access delays at lower data rates which can delay the transmission of advertisements. For example, delays up to 100 ms have been reported on 2 Mbps wireless links when the DCF mechanism is used [53]. Refer to Appendix B for further details. Furthermore, 802.11 APs may drop or further delay broadcast advertisements that originate from the wired network. When all of these factors are considered, advertisement-based movement detection may perform unpredictably when high advertisement broadcast rates are used over 802.11 wireless LANs.

6.4 Phase 1 – Handoff Latency Evaluation

6.4.1 Lazy-binding

In all the evaluation experiments, five different advertisement intervals were considered from 1 to 5 seconds. The 1 second interval is the minimum rate recommended by Mobile IPv4 RFC 2002, while 3 seconds is the IPv6 Neighbour Discovery minimum. Experiments could not be performed using intervals less than 1 second because the Dynamics software does not support these values. Intervals above 5 seconds result in excessive handoff delays which severely disturb even non-real-time applications such as FTP and SSH connections and will therefore not be considered.

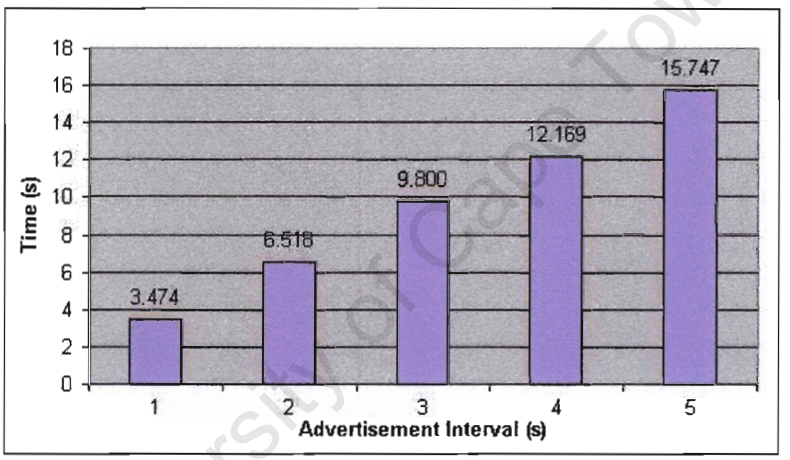

Figure 6.3: Mobile IPv4 handoff using lazy-binding

The default lazy-binding or Lazy Cell Switching (LCS) mechanism represents the worst-case movement detection performance. Figure 6.3 demonstrates the total Mobile IP handoff latencies, including 802.11 handoff delays, that are experienced for each interval when LCS is used. These values represent the time interval between the initiation of a link layer handoff and successful registration, during which the MN is unable to transmit or receive IP packets. Figure 6.3 illustrates that this movement detection technique is highly dependent on the advertisement broadcast rate and delays increase linearly with the advertisement interval. At the Mobile IPv4 (RFC 2002) minimum, the MN's traffic is subjected to delays of approximately 3.5 seconds. This is unsuitable for any type of real-time system

such as VoIP. When these Mobile IP handoff latencies are compared to the corresponding movement detection delays in Figure 6.4, it is clear that movement detection is by far the greatest contributor to the total Mobile IP handoff latency.

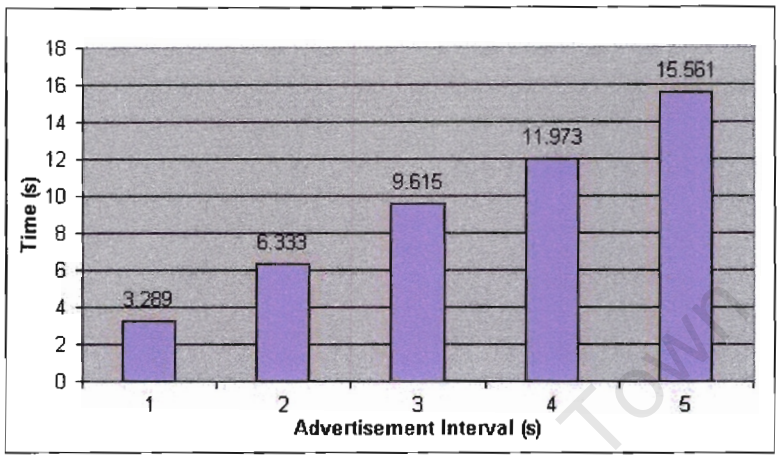

Figure 6.4: Lazy-binding movement detection delay

Table 6.1 outlines the expected values (in seconds) for agent discovery and selection using the formulae[2] presented in section 3.6.1. Figure 6.5 also focuses on the total movement detection delay and isolates the delays caused by both the agent discovery and selection phases. This figure indicates that the expected and measured agent discovery and selection delays are in accordance.

Advertisement interval (s)	Agent discovery (s)	Agent selection (s)
1	0.5	2.5
2	1	5
3	1.5	7.5
4	2	10
5	2.5	13.5

Table 6.1: Theoretical agent discovery and selection times

On average, a MN will arrive on a new IP subnet halfway through the advertisement interval. The MN will therefore have to wait for half the interval before a new advertisement

[2]Assuming for simplicity that $MaxInterval = MinInterval$.

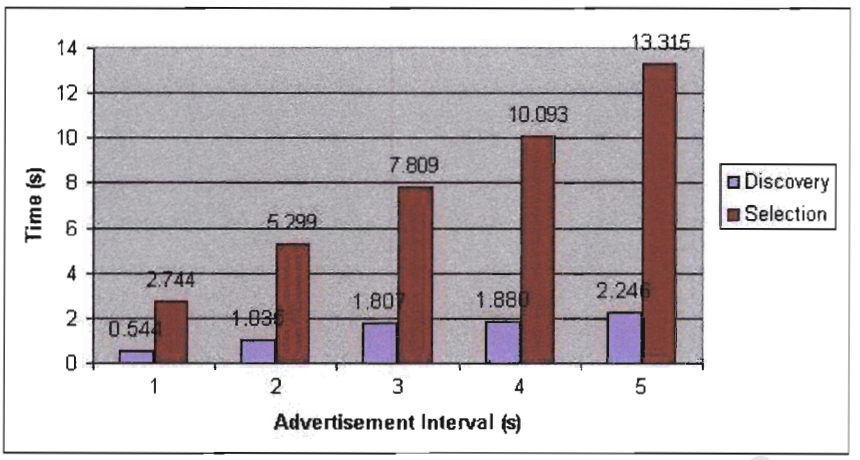

Figure 6.5: Observed agent discovery and selection phases

is received. This expected discovery delay is confirmed in Figure 6.5. However, because the arrival of a MN on a link is an independent event and a MN may arrive at any time during the interval, individual agent discovery delays should be evenly distributed over the advertisement interval. An example of this using a 1 second interval is shown in Figure 6.6.

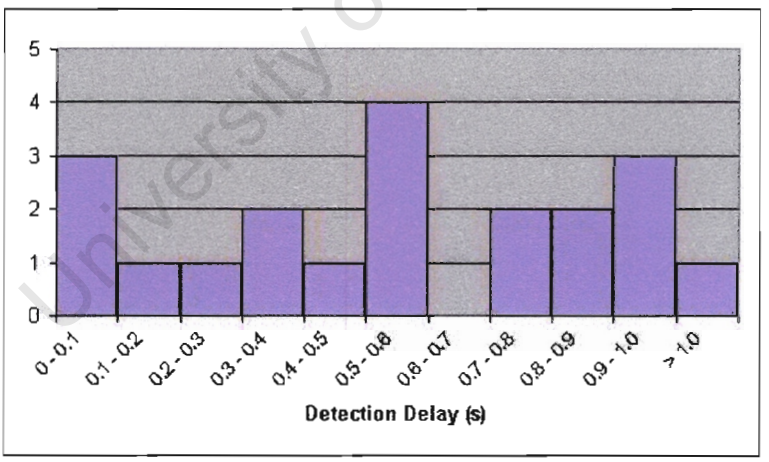

Figure 6.6: Distribution of agent discovery delays

In the second phase of LCS movement detection (agent selection), the old agent expires and the newly discovered agent is selected. Using a lazy-binding policy, this selection only occurs after three consecutive advertisements from the old agent have been missed. On

average, this takes place after 2.5 advertisement intervals have passed (see Table 6.1). The agent selection delays observed within the evaluation framework are in accordance with these predictions, as illustrated by Figure 6.5.

Figure 6.7 illustrates the registration delay distribution. Because the registration stage is independent from the movement detection technique, a much larger set of 100 results was used for analysis. Registration delays within the evaluation framework are extremely consistent as all three IP networks are physically adjacent to one another. On average, the registration process took 30 ms to complete and similar results would be expected within a micromobility framework.

Average registration delay	30 ms
Standard deviation	<1 ms

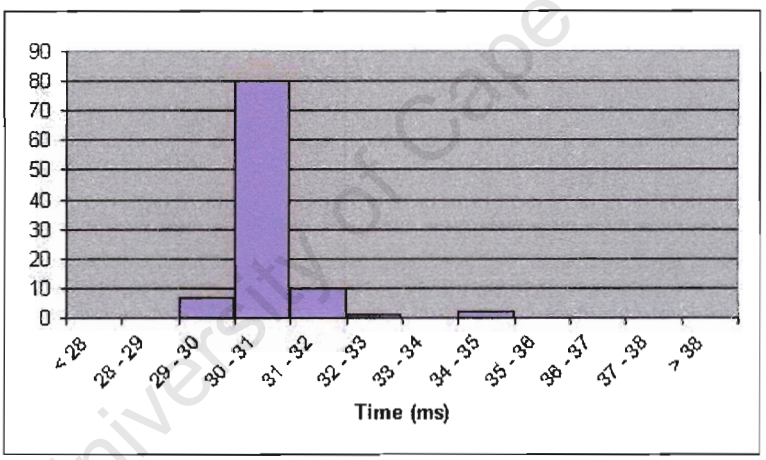

Figure 6.7: Distribution of registration delays

It is believed that the current registration delays are influenced by the limitations of the home agent. During preliminary testing, registration delays would vary considerably, sometimes reaching 80 ms. In an effort to resolve these fluctuations, several peripheral OS services were deactivated. This improved the registration performance significantly as the home agent daemon was allocated additional system resources and processor time.

6.4.2 Eager-binding

The eager-binding (or ECS) movement detection technique improves handoff performance by avoiding agent selection delays. However, the MN still relies on periodic advertisements to detect movement. Much like the lazy-binding policy, the MN must wait for half the advertisement interval (on average) before a new agent is detected. However, once the agent is detected, agent selection is performed immediately.

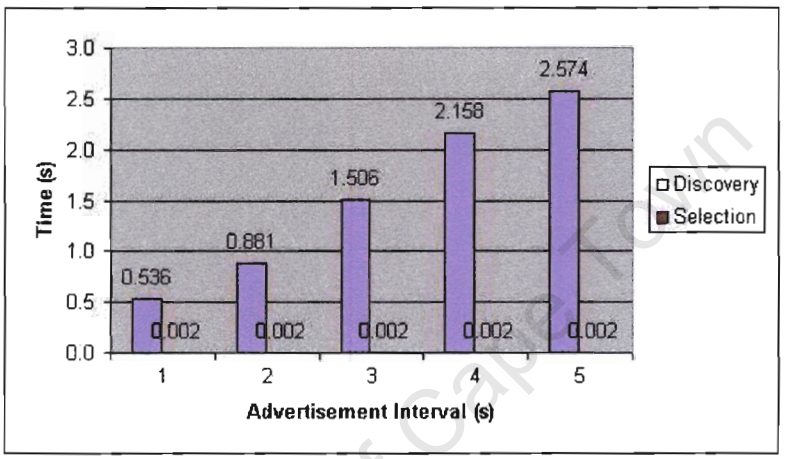

Figure 6.8: Eager-binding agent discovery and selection delays

Figure 6.9 depicts the total Mobile IP handoff latency using the eager-binding scheme. The figure indicates that the performance of this mechanism is still strongly dependent on the advertisement rate because the MN must wait for a periodic advertisement. The agent discovery delays are therefore the same as for the lazy-binding policy. However, the agent selection time is reduced below 2 ms. In the eager-binding mechanism, along with all subsequent movement detection schemes, the agent selection delay is dependent on the processing power of the MN and the extent to which the Dynamics source code has been optimised. Further speed optimisations to the Dynamics code may eliminate agent selection delays altogether.

The avoidance of agent selection delays, as shown in Figure 6.9, results in significantly faster Mobile IP handoff. Movement detection optimisations aim to further reduce these delays by removing the MN's dependence on the agent advertisement rate altogether, as discussed below.

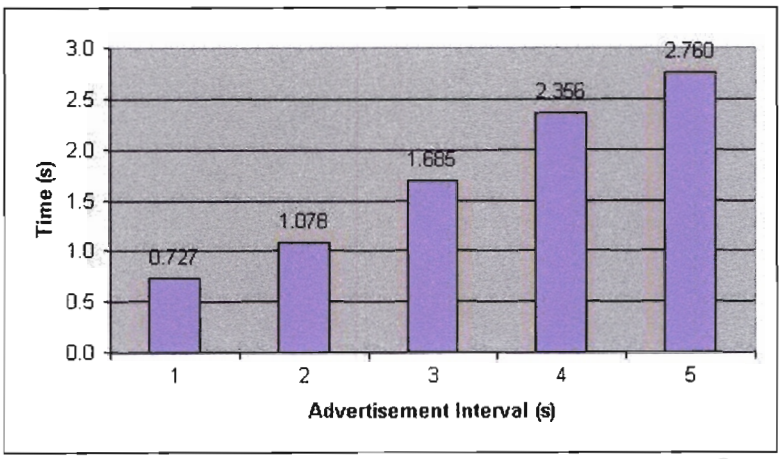

Figure 6.9: Total Mobile IP handoff latency using eager-binding

6.4.3 Hinted Cell Switching

In the Hinted Cell Switching (HCS) mechanism, the MN broadcasts a solicitation after an 802.11 handoff and receives a randomly delayed advertisement in reply. Figure 6.10 outlines the agent discovery and selection delays for this optimisation. An eager-binding selection policy is used which essentially eliminates the selection phase. The time until a new agent is discovered through an advertisement after arrival on a new link has also been reduced dramatically. Individual agent discovery times vary randomly between 4 ms and 200 ms. This is due to the random delay that mobility agents insert before responding to solicitations[3].

As seen in Figure 6.11, a combination of the HCS and FastADV techniques results in a further performance improvement. When the random delay is removed for solicitation/advertisement exchanges, replied advertisements allow the MN to discover new agents within a shorter time. The FastADV mechanism results in approximately 80-90% faster agent discovery. However, agent discovery delays are introduced by the finite time required to transmit a solicitation and receive a replied advertisement over the wireless medium, along with the processing of the solicitation by the fastadv process.

When combined HCS/FastADV movement detection is performed, the total Mobile IP handoff latency is reduced to approximately 200 ms. In contrast with the default lazy/eager

[3]A Dynamics MN uses a smaller random delays than the values recommended by MIPv4. Refer to section 3.3.1 for details.

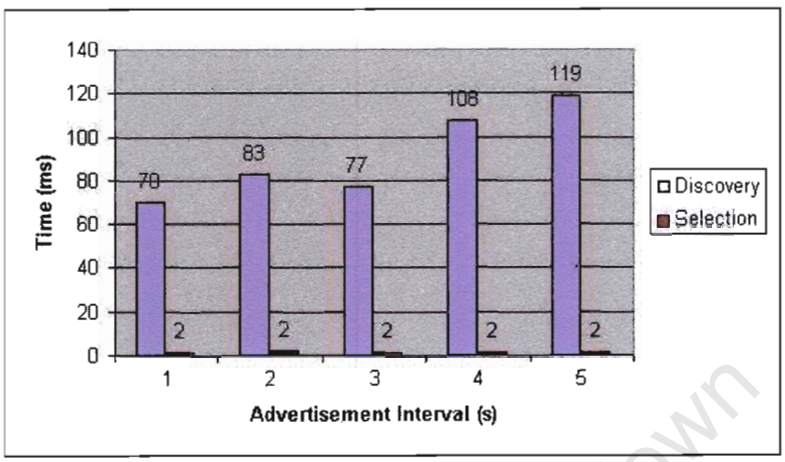

Figure 6.10: HCS agent discovery and selection delays

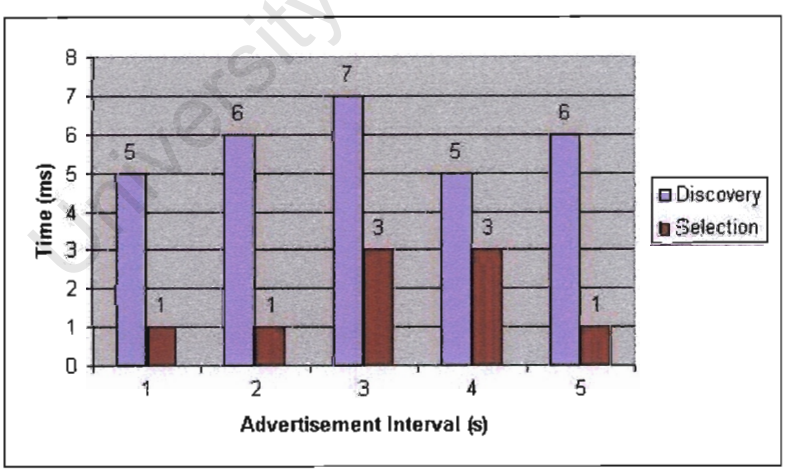

Figure 6.11: HCS and FastADV agent discovery and selection delays

schemes, the main component of a Mobile IP handoff is no longer movement detection but 802.11 handoff. Figure 6.12 illustrates that Mobile IP handoff is also no longer directly related to the advertisement rate.

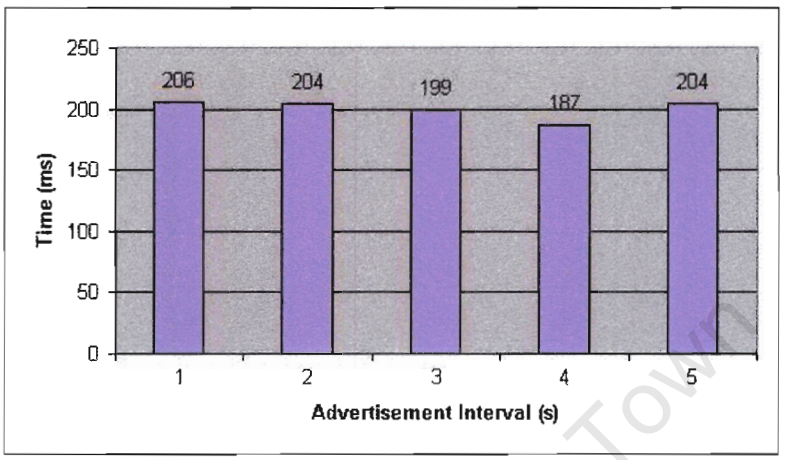

Figure 6.12: Total Mobile IP handoff latency using HCS (with FastADV)

6.4.4 Advertisement Caching

The agent discovery and selection delays for the advertisement caching technique are depicted in Figure 6.13. This technique results in the lowest agent discovery delays of all the evaluated mechanisms, which in turn produces the best movement detection performance. The advertisement caching mechanism ensures that the delays caused by solicitation/advertisement exchanges are avoided. However, Figures 6.11 and 6.12 illustrate that the HCS/FastADV combination provides an effective fall-back mechanism if a cached advertisement is missed.

As mentioned previously, agent selection is executed by the Dynamics software and delays are introduced by the processing time of the software itself. The Dynamics package is currently prototype software and has not been optimised for efficiency or speed. This factor introduces the agent selection delay variations seen in Figure 6.13.

As with the previous HCS mechanism, the Mobile IP handoff latency consists almost entirely of 802.11 handoff delay (Figure 6.14). Although the advertisement caching scheme results in the lowest movement detection delay, the performance gains of this technique

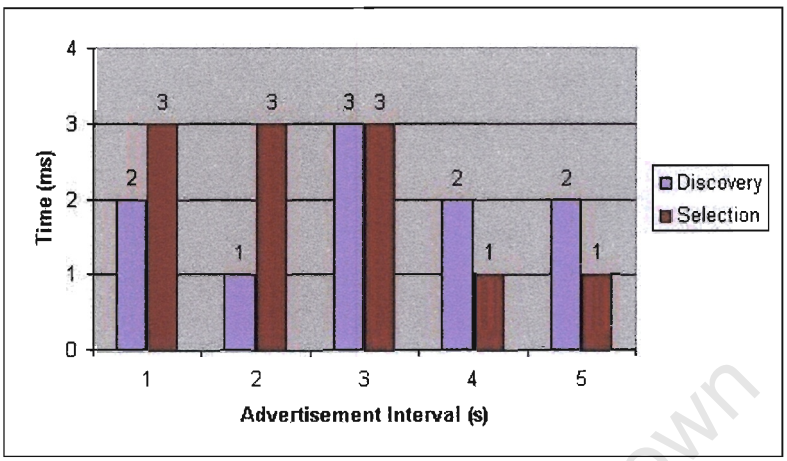

Figure 6.13: Advertisement caching agent discovery and selection delays

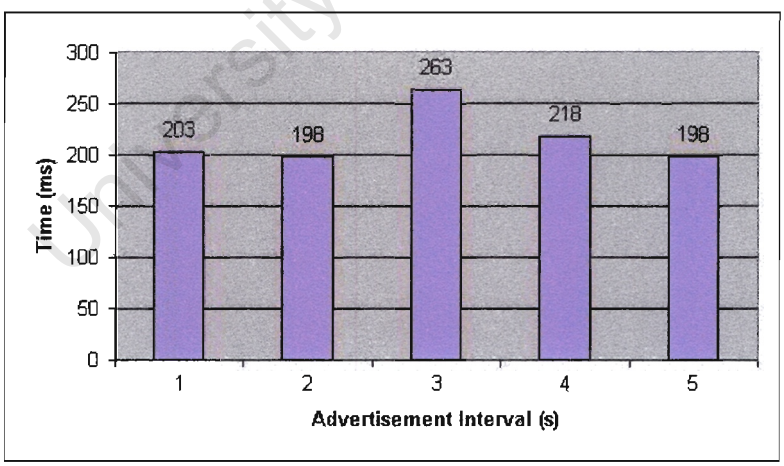

Figure 6.14: Total Mobile IP handoff latency using advertisement caching

as compared with HCS are not apparent when the total Mobile IP handoff latency is considered. The reason for this is that 802.11 handoff delays are relatively large and hide these movement detection improvements. In addition, the variations in the total Mobile IP handoff results depicted in Figures 6.12 and 6.14 are mainly due to the large variations in the 802.11 handoff process.

6.5 Measurement of Speech Quality

The performance of a VoIP system during a Mobile IP handoff is evaluated by subjectively assessing the output voice quality. The guidelines for how this assessment should be performed are based on ITU-T recommendation P.800 [49]. The ITU-T recommendation describes different techniques that can be used to subjectively evaluate a voice transmission system that introduces distortions such as packet loss or transmission errors. This recommendation provides very detailed guidelines regarding the conditions, requirements and procedures for performing these assessments. As the measurement of speech quality in the event of packet loss is not the main focus of this study, a very rudimentary evaluation will be used to illustrate the vulnerabilities of a VoIP system during a Mobile IP handoff. Consequently, not all of the recommendations described in the ITU-T document have been adhered to.

A subjective analysis of voice quality can be performed either using a conversational-opinion or listening-opinion test. Conversational opinion methods aim to reproduce scenarios that would be experienced by users. For example, two users placed in separate sound-proof rooms are asked to assess the quality of their conversation.

Listening-opinion tests are not as accurate as conversational-opinion methods because they rely on more artificial scenarios. In this case, voice information flows in only one direction (e.g. a user passively listening to a speech sample passed through the transmission system). These methods can be applied to two-way systems such as VoIP, but are not as comprehensive as conversational-opinion tests. In spite of these shortcomings, this method was used to evaluate the output voice quality during a Mobile IP handoff because it is much less complicated to implement and the requirements are less stringent. Refer to the ITU recommendation for further details [49].

Listening-opinion tests are carried out in the form of an Absolute Category Rating (ACR) survey whereby users are played a speech sample that has been subjected to certain trans-

mission effects, such as a Mobile IP handoff. The listeners[4] then judge the speech quality according to the scale shown in Table 6.2 [49]. Each listener is asked to rate the sample quality with specific consideration to the information lost during a handoff. The opinion results of each sample are then averaged to form the mean opinion score (MOS).

Speech Quality	Score
Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

Table 6.2: Listening quality scale

The ITU-T recommendation specifies several parameters and restrictions that the tests should comply with. For example, the speech sample should consist of short sentences and the ambient noise in the listening environment should meet certain criteria. However, many of these restrictions will not be enforced because the listening-opinion scores will only be used to provide a very basic assessment of Mobile IP handoff. This is especially applicable to the FastADV and advertisement caching techniques where the improvements in movement detection are completely masked by the large 802.11 handoff delays.

6.6 Phase 2 – VoIP Performance Evaluation

6.6.1 Packet Loss

Table 6.3 describes the average number of VoIP packets lost during a Mobile IP handoff using different movement detection methods. A one second advertisement interval was used throughout these tests. This table illustrates the direct relationship that exists between the handoff latency and packet loss. VoIP packets were transmitted through the emulation framework at a rate of 5 packets per second and therefore these results agree with the handoff latency values presented above.

[4] 15 listeners have been surveyed, which is in accordance with the ITU recommendation [49].

Movement Detection Technique	Average Number of Packets Lost
Lazy-binding (LCS)	159
Eager-binding (ECS)	42
HCS	12
HCS/FastADV	9
Advertisement caching	9

Table 6.3: Average packet loss during a Mobile IP handoff

The results listed in Table 6.3 assume that the PCM G.711 codec is used. Different codecs produce different data rates and thus the packet loss and corresponding data loss during a Mobile IP handoff both depend on the characteristics of the particular codec.

Another observation that was made during these tests is that both VoIP jitter and delay requirements are satisfied during Mobile IP handoffs. This means that packets before and after handoff are not subjected to excessive delays or jitter. The only quantity that is significantly affected is packet loss.

6.6.2 Subjective Quality Assessment

Five samples were played to each listener where each sample contained a segment of information loss due to a Mobile IP handoff. As shown below in Table 6.4, every sample contained an artifact introduced by a Mobile IP handoff using a specific movement detection technique. Table 6.4 indicates the average user opinion of each sample.

Movement Detection Technique	Sample	MOS
Lazy-binding	A	1.3
Eager-binding	B	1.3
HCS	C	3.1
HCS with FastADV	D	3.3
Advertisement caching	E	3.6

Table 6.4: Listener opinion scores

These MOS values only present a very general indication of each sample's quality. During these evaluations, emphasis has been placed on establishing a trend in user opinions

rather than the numerical accuracy of these results. It is important to note therefore that these results are not completely reliable, due to the simplifications mentioned previously. Furthermore, the variation in 802.11 handoff delays and the position of the loss within the sample decrease the survey's accuracy. However, these MOS values can still be used, for the purposes of this study, to perform a rough analysis that illustrates general concepts[5]. Significant further work is required in this area before the output voice quality can be measured with a high degree of accuracy.

Table 6.4 indicates that there is a clear distinction between the default Mobile IP movement detection techniques and the optimised mechanisms. Despite the significant reduction in handoff delays induced by the eager-binding policy, most users found the resulting loss of information to be unacceptable in both the eager and lazy policies. On the other hand, the three movement detection optimisations all resulted in MOS values that reflected "fair" to "good" voice quality. This can be interpreted to mean that although the loss of information could be audibly detected, most listeners believed that this did not completely degrade the voice quality.

These MOS values agree with the packet loss and delay results discussed previously. For example, handoff delays are significantly reduced when HCS is used as compared with ECS (approximately 70% improvement). These delay and packet loss reductions result in a significantly higher output voice quality (from "bad" to "fair"). However, the MOS of the last three samples are roughly the same, despite the fact that FastADV and advertisement caching improve movement detection delays by 80-90% over HCS. The reason for this is that the marginal improvements that FastADV and advertisement caching introduce are relatively small as compared to the large and variable 802.11 handoff latencies. As a result, listeners are generally unable to detect these differences. The reason for the slightly higher MOS associated with advertisement caching is not due to its movement detection performance. Rather, it was found that the 802.11 handoff in sample E was significantly shorter than the other samples (130 ms as opposed to approximately 200 ms).

6.7 Hybrid Technique

The hybrid mechanism relies on the movement detection optimisations discussed above to deliver new advertisements. The characteristics of these mechanisms (including latency

[5]For this reason, a full statistical analysis (including significance tests) has not been performed

and packet loss) have been already been presented and will not be revisited in this section. Rather, the monitoring and policy selection functions[6] of the hybrid system are evaluated below. The MN monitor logs all relevant input information and policy decisions to a log file that can be analysed after the tests have been completed. The informative sections of this log file will be presented and discussed in this section.

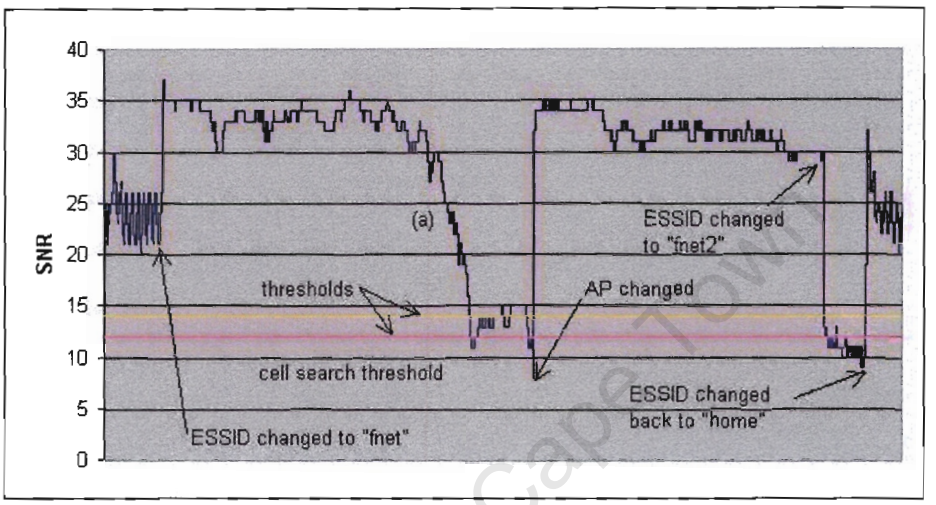

Table 6.5: SNR log of MN movement

In order to evaluate the hybrid system, the MN performed both inter-ESS 802.11 handoffs between different IP networks and intra-ESS 802.11 handoffs within the same IP network. Figure 6.5 illustrates the SNR history logged by the MN monitor as the MN performed these different handoffs. Inter-ESS handoffs were performed by changing the ESSID as mentioned previously. In Figure 6.5, the MN moved from its home network through two different foreign networks (with ESSIDs "fnet" and "fnet2" respectively) and back to its home network.

The "fnet" IP network contains two foreign agents/APs. While the MN was attached to this network, an intra-ESS was performed between the two APs. Intra-ESS handoffs are usually initiated by a decline in SNR which typically occurs as a MN moves further and further from its current AP. As the MN was implemented on a fixed desktop computer, this behaviour was emulated by progressively blocking the AP's radio signal using a physical obstacle (such as a sheet of metal). This caused the SNR of the MN's received signal

[6]refer to Figure 5.5

to decrease, as indicated in Figure 6.5 (a). The policy thresholds (including a region of hysteresis) displayed in Figure 6.5 were used to proactively select the eager-binding policy in response to the decrease in SNR. When the SNR fell below the *cell search threshold*, an intra-ESS 802.11 handoff was initiated to a neighbouring AP.

The following extract was taken from the MN monitor log file during movement between the wireless LAN on the home network ("home") and the "fnet" wireless LAN located on a foreign network.

```
ESSID:"home"
AP: 00:0D:88:50:70:17    SNR: 26
--------------------------------------------------
ESSID:"fnet"
AP: 44:44:44:44:44:44    [Invalid AP]     SNR: 26                              (1)
AP connection lost: activating ECS
Kernel Info: Dec 9 15:02:21 Photon kernel: eth1: New link status: Disconnected      (2)
--------------------------------------------------
ESSID:"fnet"
AP: 44:44:44:44:44:44    [Invalid AP]     SNR: 26
--------------------------------------------------
[CUT][7]
--------------------------------------------------
ESSID:"fnet"
AP: 44:44:44:44:44:44    [Invalid AP]     SNR: 26
--------------------------------------------------
ESSID:"fnet"
AP: 00:60:1D:1F:33:11    SNR: 26
New AP detected.                                                              (3)
New network. Old: 00:0D:88:50:70:17 New: 00:60:1D:1F:33:11
ECS maintained.
--------------------------------------------------
ESSID:"fnet"
AP: 00:60:1D:1F:33:11    SNR: 26
--------------------------------------------------
ESSID:"fnet"
AP: 00:60:1D:1F:33:11    SNR: 26
Kernel Message: Dec 9 15:02:21 Photon kernel: eth1: New link status: Connected      (4)
--------------------------------------------------
ESSID:"fnet"
AP: 00:60:1D:1F:33:11    SNR: 37
Counter started...
```

The MN monitor ascertains when the link to its associated AP has been lost by detecting both the invalid AP MAC address (1) and receiving kernel information from the syslog interface (2). This implies that an inter-ESS handoff has been initiated and the ECS

[7] [CUT] indicates that sections have been edited out to save space.

policy is activated in response. When a valid AP MAC address is eventually detected, the MN monitor verifies whether the new AP is on a different IP network. Ideally this would be achieved using a mechanism such as IAPP but for simplicity, this information was hard-coded within the MN monitor source code. At this stage (3), the MN's wireless card has still not completed the link layer handoff and is executing the authentication or association services. 802.11 handoff is completed only a few milliseconds later, when a link-up hint is received via the syslog interface (4). A counter is then started to ensure that the ECS policy remains active for a certain time period after an 802.11 handoff or until a new cached advertisement is received. If no advertisement is received within this time[8], the MN falls back on the HCS/FastADV policy and subsequently deactivates its eager-binding.

The extract below was produced when an intra-ESS handoff was performed within the "fnet" foreign network. This happened when the "AP change" occurs after region (a) in Figure 6.5.

```
ESSID:"fnet"
AP: 00:60:1D:1F:33:11    SNR: 11
Lower Threshold passed: activating ECS                                    (1)
--------------------------------------------------
[CUT]
--------------------------------------------------
ESSID:"fnet"
AP: 00:60:1D:1F:33:11    SNR: 9
--------------------------------------------------
ESSID:"fnet"
AP: 00:02:2D:01:3C:BE    SNR: 8
New AP detected (Reassociation).                                          (2)
AP on same network. Old: 00:60:1D:1F:33:11 New: 00:02:2D:01:3C:BE
Activating LCS.
--------------------------------------------------
[CUT]
--------------------------------------------------
ESSID:"fnet"
AP: 00:02:2D:01:3C:BE    SNR: 8                                           (3)
Kernel Info: Dec 9 15:03:13 Photon kernel: eth1: New link status: AP Changed
--------------------------------------------------
ESSID:"fnet"
AP: 00:02:2D:01:3C:BE    SNR: 32                                          (4)
```

In this case, the ECS policy is activated when the SNR falls below the lower policy threshold (1). The MN monitor then detects an 802.11 handoff through a change in the current

[8] 30 ms has been assumed in all experiments. However, further study is needed to determine the most appropriate value for this delay.

AP MAC address and verifies that the new AP resides on the current IP subnet (2). The monitor subsequently enforces the LCS policy (ignoring SNR) until a kernel indication is received that 802.11 handoff has been completed. Once this indication is received, the policy selection continues as normal (based on SNR) (3). The reason for enforcing LCS until 802.11 handoff is over is to prevent the ECS policy from being activated by an inaccurate SNR value, shown in (3) above. The SNR value at this stage has not been updated, which is apparent when values (3) and (4) are compared.

The hybrid system evaluations that were carried out on the modified evaluation framework exposed two interesting characteristics about 802.11 handoffs. These characteristics are illustrated within the log file extracts above.

- Firstly, when an 802.11 performs an inter/intra-ESS handoff, the MAC address of the new AP is available before the handoff is completed. The detection of the new AP MAC address occurs once the scanning phase has been completed and the new AP selected. However, an operational link has not yet been established at this stage because the authentication and association phases are still underway. The SNR during this period is not updated and reflects the previous link's quality. A kernel indication is received once the new wireless link is finally active. Notice that the SNR is only updated to reflect the new link's quality at this point. This factor is extremely useful to the hybrid system because it ensures that the policy selection is executed immediately before the MN establishes a new link. In other words, this ensures the MN arrives on a link with the correct policy in place (LCS/ECS).

- Another interesting characteristic of 802.11 handoffs is the difference between intra and inter-ESS handoff latencies. The results in section 6.2 show that 802.11 handoffs between different wireless LANs last 156 ms on average, and this is confirmed by the first extract above. However, the second extract illustrates that intra-ESS handoffs introduce significantly lower delays (approximately 40-50 ms). From these results it is clear that an 802.11 station is able to simplify its handoff procedure when reassociating with an AP that is part of its ESS. However, it remains unclear how this lower latency is achieved in practice.

Chapter 7

Conclusions

The main objective of this study is the investigation of different mechanisms and architectures that improve Mobile IP movement detection performance. Movement detection optimisations aim to reduce both the latency and packet loss of Mobile IP handoffs by ensuring that IP layer movement is detected as quickly as possible. The basis of these optimisations is that information from the link layer can be used to attain these goals.

A network architecture model has been developed throughout the previous chapters that provides wireless network access to mobile users. This framework combines both 802.11 wireless technology and Mobile IP mobility management. This study has demonstrated, both analytically and practically, that 802.11 link layer information can be used effectively to improve Mobile IP handoff performance.

Both Mobile IP versions 4 and 6 have been described and compared. Although the Mobile IPv6 protocol includes many enhancements and is better integrated into IPv6, it does not currently provide any direct gains with regard to handoff performance. However, the recent developments within the "Detecting Network Attachment" working group are of direct relevance to Mobile IPv6. In contrast, Mobile IPv4 handoff advancement has effectively come to an end. As a result, future progress in the area of Mobile IP handoff will most likely occur within the context of IPv6.

Mobile IP handoffs and their components have been described in detail in the previous chapters. In addition, the 802.11 handoff process, along with its contributions to Mobile IP handoff latency, has been investigated. It was found that the 802.11 link layer significantly affects the operation of Mobile IP in a number of different ways. In fact, seamless Mobile IP handoffs are almost impossible to achieve using current 802.11 devices.

A Mobile IP evaluation framework, based on the network model described above, has been successfully implemented using the Dynamics Mobile IPv4 project. This network has allowed Mobile IP handoffs to be analysed in detail, with specific focus on the properties of movement detection. The characteristics of 802.11 handoff and Mobile IP registration procedures that further contribute to the total handoff latency have also been assessed. Despite its limitations, the evaluation network has supported the successful deployment and testing of several different movement detection mechanisms. Furthermore, the extent to which different movement detection optimisations improve Mobile IP handoff latency has been determined using this framework. This analysis has been performed using a subjective assessment of a VoIP application, in addition to the numerical evaluation of both handoff latency and packet loss. The following conclusions have been drawn from experiments performed using the evaluation framework:

- 802.11 handoffs between different wireless LANs introduce significant delays into the Mobile IP handoff process. These findings are consistent with the results reported in previous studies, as described in Chapter 2. These link layer delays are unavoidable in commercial wireless LAN cards as link layer processes are hidden within the firmware/hardware of these devices.

- On the other hand, terminal movement within a single wireless LAN was discovered to result in much lower link layer handoff delays. The exact reasons for these differences have not been determined.

- The evaluation framework's structure, where all IP networks are in relative physical proximity, emulates a micromobility architecture. This demonstrates that registration delays can be minimised effectively by ensuring that registration message exchanges are executed within a local administrative domain instead of through the Internet.

- The default Mobile IP movement detection policies introduce significant interruptions into the MN's upper-layer sessions during a handoff because they rely exclusively on periodic advertisements. These results are confirmed by the fact that the output VoIP quality was considered unsatisfactory for both of these policies. As a result, pure advertisement-based movement detection is unsuitable for supporting real-time applications such as VoIP.

- The movement detection optimisations evaluated in this study dramatically improve both movement detection and Mobile IP handoff performance as compared with the default Mobile IP mechanisms. All of these techniques result in adequate VoIP output quality. The main factor that contributes to packet loss and the subsequent degradation in VoIP quality is the 802.11 handoff delay. This is in contrast with the default Mobile IP mechanisms, where movement detection is the most time consuming process.

- The above results imply that the 802.11 handoff process will have to be optimised before any significant further improvements to Mobile IP handoff performance are possible. These findings confirm the results published by the independent studies presented in Chapter 3.

The proposed hybrid mechanism was successfully implemented and tested on the evaluation framework. The hybrid selection policy performs robustly while the MN moves within a wireless LAN containing more than one foreign agent. Movement detection policies are activated appropriately whether a MN roams within a wireless LAN or migrates between two adjacent wireless LANs on separate IP subnets.

The evaluation framework has shown that hint-based movement detection optimisations are an effective way of reducing Mobile IP handoff latency. Many of these optimisations will ensure that real-time VoIP communications are not significantly affected by Mobile IP handoffs, provided that 802.11 handoff delays are minimised. Furthermore, evaluations using the hybrid system have demonstrated that a combination of movement detection techniques both improves Mobile IP handoff performance and increases the overall reliability of a Mobile IP system.

Chapter 8

Recommendations

This study has encompassed a broad spectrum of networking technologies. These range from 802.11 which deals with local connectivity to VoIP which is a real-time, end-to-end application. As these different technologies were investigated, a number of avenues for further research became evident. Listed below is a brief outline of some of the most important recommendations that came to light during the course of this project.

- The limitations of current commercial 802.11 devices have been illustrated in detail. Further research is needed to improve the performance of 802.11 handoffs. The discussions taking place within the IEEE 802.21 Handover working group indicate that this promises to be an active area of research in the near future. Furthermore, the improvement of support for real-time applications such as VoIP and the introduction of quality of service (QoS) in wireless LANs are also important issues that are being investigated within the 802.11e working group.

- Other wireless LAN technologies such as 802.11a and HIPERLAN/2 must be investigated to determine whether they offer improved handoff performance as compared to 802.11b used in this study.

- This study performed an elementary evaluation of the effects that Mobile IP handoffs have on VoIP call quality. Additional in-depth research using both subjective and objective techniques must be carried out in order to extend these results.

- The evaluation framework has been designed to support the addition of various extensions to the existing infrastructure. A larger and more realistic evaluation framework

would be useful in testing a broader range of applications (e.g. streaming video). It would also aid in determining the scalability of the techniques discussed in this study when a larger network containing a greater number of mobile nodes is considered.

- Additional work is needed to develop and test the hybrid system further within a larger and more intricate framework. The hybrid system should also be evaluated using "real" mobility scenarios (e.g. public hot-spot or office environment). In addition, the interaction between the hybrid system and IAPP must be developed further.

- Security issues are of great concern within the context of network mobility and have largely been ignored in this study. Further work must be devoted to the security mechanisms of both 802.11 wireless LANs and Mobile IP. For example, efficient ways of authenticating users as they migrate through different networks (e.g. through enhanced AAA[1] mechanisms) must be developed if practical wireless access networks are to be feasible.

- Mechanisms that perform movement/network detection within the context of IPv6 are currently being developed within the IETF "Detecting Network Attachment" (DNA) working group, along with other related issues. Several proposals have been raised in this forum that aim to streamline IPv6 network detection. Only some of these have been discussed in this study. Further issues, such as optimised DAD (duplicate address detection), are currently under development and have direct relevance to Mobile IP movement detection. This promises to be another dynamic area of research.

[1] Authentication, Authorisation and Accounting

Bibliography

[1] HermesAP. [Online] Available: `http://hunz.org/hermesap.html`. Accessed 2004.

[2] Linux Ethernet Bridging. [Online] Available: `http://bridge.sourceforge.net`. Accessed 2004.

[3] Roaming with WaveLAN/IEEE 802.11. Wave-LAN Technical Bulletin 021/A, Lucent Technologies, 1998. [Online] Available: `http://www.orinocowireless.com/support/techbulletins/TB-021.pdf`.

[4] Planning Large Scale Installations. Orinoco Technical Bulletin 023/B, Agere Systems, April 1999. [Online] Available: `http://www.orinocowireless.com/support/techbulletins/TB-023.pdf`.

[5] Inter Access Point Protocol (IAPP). Orinoco Technical Bulletin 034/A, Agere Systems, 2000. [Online] Available: `http://www.orinocowireless.com/support/techbulletins/TB-034.pdf`.

[6] Dynamics Mobile IP. [Online] Available: `http://dynamics.sourceforge.net/`, Accessed 2004.

[7] Ebtables. [Online] Available: `http://ebtables.sourceforge.net/`, Accessed 2004.

[8] Ethereal: A Network Protocol Analyzer. [Online] Available: `http://www.ethereal.com/`, Accessed 2004.

[9] IEEE 802.21 Working Group. [Online] Available: `http://www.ieee802.org/21/`, accessed 2004.

[10] A. Aliman and B. Aboba. Analysis of Roaming Techniques. *IEEE 802.11-04/0377r1*, 2004. [Online] Available: `http://www.drizzle.com/~aboba/IEEE/`.

[11] T. W. Andersen and A. Lildballe. Seamless Handoff in Mobile IPv6. Master's thesis, Aalborg University, June 2001.

[12] B. Andersson. Dynamics - Functional Definition. Technical report, Helsinki University of Technology, April 1999.

[13] M. ILyas B. Furht, editor. *Wireless Internet Handbook*. CRC Press, 2003.

[14] M. Bandai and I. Sasase. A Low Latency Handoff Scheme Using Positional Information for Mobile IP Based Networks. *Proceedings of IEEE Globecom*, 2003.

[15] C. Castelluccia. HMIPv6: A Hierarchical Mobile IPv6 Proposal. *ACM Mobile Computing and Communication Review (MC2R)*, April 2000.

[16] J. Choi and D. Shin. Fast Router Discovery with AP Notification. *IETF Internet Draft*, June 2002. draft-jinchoi-l2trigger-fastrd-01.

[17] J. Choi and D. Shin. Fast Router Discovery with RA Caching in AP. *IETF Internet Draft*, February 2003. draft-jinchoi-mobileip-frd-00.

[18] P. De Cleyn, N. Van den Wijngaert, L. Cerda, and C. Blondia. A Smooth Handoff Scheme using IEEE 802.11 Triggers - Design and Implementation. *Computer Networks*, April 2004.

[19] T. Cornall and B. Pentland. Layer 2 Triggers Improve Handover in Wireless Mobile IPv6. [Online] Available: `http://www.telecommunications.crc.org.au/content/ConfPapers/Cornall.pdf`.

[20] G. Daley and J. Choi. Movement Detection Optimization in Mobile IPv6. *IETF Internet Draft*, May 2003. draft-daley-mobileip-movedetect-01.

[21] G. Daley, B. Pentland, and R. Nelson. Effects of Fast Router Advertisement on Mobile IPv6 Handovers. *Eighth IEEE International Symposium on Computers and Communications*, 2003.

[22] G. Daley, B. Pentland, and E. Nordmark. Deterministic Fast Router Advertisement Configuration. *IETF Internet Draft*, July 2004. draft-daley-dna-det-fastra-00.

[23] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 1883*, December 1995.

[24] A. Yegin (Editor). Supporting Optimized Handover for IP Mobility - Requirements for Underlying Systems. *IETF Internet Draft*, December 2002. draft-manyfolks-l2-mobilereq-02.

[25] A. Yegin (Editor). Link-layer Event Notifications for Detecting Network Attachments. *IETF Internet Draft*, September 2004. draft-ietf-dna-link-information-00.

[26] A. Yegin (Editor). Link-layer Triggers and Hints for Detecting Network Attachments. *IETF Internet Draft*, February 2004. draft-yegin-dna-l2-hints-01.

[27] C. Perkins (Editor). IP Mobility Support. *RFC 2002*, October 1996.

[28] C. Perkins (Editor). IP Mobility Support for IPv4. *RFC 3344*, August 2002.

[29] K. Malki (Editor). Low Latency Handoffs in Mobile IPv4. *IETF Internet Draft*, January 2004. draft-ietf-mobileip-lowlatency-handoffs-v4-08.

[30] Linux Journal Editor. Letters to the Editor. *Linux Journal*, 1998(45), January 1998.

[31] R. Koodli (Editor). Fast Handovers for Mobile IPv6. *IETF Internet Draft*, January 2004. draft-ietf-mipshop-fast-mipv6-01.txt.

[32] S. Deering (Editor). ICMP Router Discovery Messages. *RFC 1256*, September 1991.

[33] M. Ergen. IEEE 802.11 Tutorial. *University of California Berkeley*, June 2002. [Online] Available: `http://citeseer.ist.psu.edu/536785.html`.

[34] A. Singh et al. Fast Handoff L2 Trigger API. *IETF Internet Draft*, October 2002. draft-singh-l2trigger-api-00.

[35] N. A. Fikouras and C. Görg. A Complete Comparison of Algorithms for Mobile IP Hand-offs with Complex Movement Patterns and Internet Audio. *Proceedings of the Fourth International Symposium on Wireless Personal Multimedia Communications (WPMC)*, September 2001.

[36] N. A. Fikouras and C. Görg. Performance Comparison of Hinted and Advertisement Based Movement Detection Methods for Mobile IP Hand-offs. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 37(1):55–62, 2001.

[37] N. A. Fikouras, A. J. Konsgen, and C. Görg. Accelerating Mobile IP Hand-offs through Link-layer Information, An Experimental Investigation with 802.11b and Internet Audio. *Proceedings of the International Multiconference on Measurement, Modelling, and Evaluation of Computer-Communication Systems (MMB)*, September 2001.

[38] A. Fladenmuller and R. De Silva. The Effect of Mobile IP Handoffs on the Performance of TCP. *Mobile Networks and Applications*, 4:131–135, 1999.

[39] E. Flynn. Wireless in the World. *Newsweek*, June 7/June 14 2004.

[40] P. Garg, R. Doshi, R. Greene, M. Baker, M. Malek, and X. Cheng. Using IEEE 802.11e MAC for QoS over Wireless. *The Proceedings of the 22nd IEEE International Performance Computing and Communications Conference (IPCCC 2003)*, 2003.

[41] 802.21 Working Group. IEEE-SA Standards Board Project Authorization Request Form. 2002. [Online] Available: `http://www.ieee802.org/21/802_21_PAR.doc`.

[42] 802.21 Working Group. Criteria for Standards Development (Five Criteria). 2003. [Online] Available: `http://www.ieee802.org/21/802_215Criteria.doc`.

[43] V. Gupta and D. Johnston. A Generalized Model for Link Layer Triggers. *IEEE 802.21 Working Group*, March 2004. [Online] Available: `http://www.ieee802.org/handoff/march04_meeting_docs/Generalized_triggers-02.pdf`.

[44] D. Hole and F. Tobagi. Capacity of an IEEE 802.11b Wireless LAN supporting VoIP. *IEEE International Conference on Communications (ICC)*, 2004.

[45] IEEE/ANSI. Higher-Speed Physical Layer Extension in the 2.4 GHz Band. *IEEE Standard*, 1999.

[46] IEEE/ANSI. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Standard*, 1999.

[47] IEEE/ANSI. Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band. *IEEE Standard*, 2003.

[48] IEEE/ANSI. IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation. *IEEE Standard*, June 2003.

[49] ITU-T. Methods for Subjective Determination of Transmission Quality. August 1996. ITU-T Recommendation P.800.

[50] ITU-T. Transmission Systems and Media, Digital Systems and Networks: One Way Transmission Time. 2003. ITU-TG114.

[51] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. *RFC 3775*, June 2004.

[52] J. Kempf, M. Khalil, and B. Pentland. IPv6 Fast Router Advertisement. *IETF Internet Draft*, March 2002. draft-mkhalil-ipv6-fastra-03.

[53] A. Kopsel and A. Wolisz. Voice Transmission in an IEEE 802.11 WLAN Based Access Network. *International Workshop on Wireless Mobile Multimedia*, 2002.

[54] S. Loosemore. *The GNU C Library Reference Manual*, August 1999.

[55] P. Mehta. Overview of Voice over IP. *Technical Report MS-CIS-01-31*, 2001. [Online] Available: `http://www.cis.upenn.edu/~udani/papers/OverviewVoIP.pdf`.

[56] D. Miras. A Survey of QoS Needs of Advanced Internet Applications. 2002. [Online] Available: `http://qos.internet2.edu/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.pdf`.

[57] A. Mishra, M. Shin, and W. Arbaugh. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. *ACM SIGCOMM Computer Communication Review*, 33(2), April 2003.

[58] A. Mishra, M. Shin, and W. Arbaugh. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. *IEEE INFOCOM*, 2004.

[59] N. Montavont and T. Noel. Analysis and Evaluation of Mobile IPv6 Handovers over Wireless LAN. *Mobile Networks and Applications*, 8(6):643 – 653, December 2003.

[60] S. Narayanan, G. Daley, and N. Montavont. Detecting Network Attachment in IPv6 - Best Current Practices. *IETF Internet Draft*, October 2004. draft-narayanan-dna-bcp-01.

[61] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). *RFC 2461*, December 1998.

[62] T. Narten and S. Thomson. IPv6 Stateless Address Autoconfiguration. *RFC 2462*, December 1998.

[63] UCL (University College London) Network and Multimedia Research Group. The Robust Audio Tool (RAT). [Online] Available: `http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/`, Accessed 2004.

[64] S. Daniel Park. Minimized Duplicate Address Detection. *IETF Internet Draft*, September 2003. draft-park-dna-mdad-00.

[65] S. Daniel Park. Requirement for IPv6 DAD Optimization. *IETF Internet Draft*, July 2003. draft-park-ipv6-optidad-requirement-01.

[66] S. Daniel Park and S. Joo Koh. Fast Handover Agent (FHA) for Fast Router Discovery in FMIPv6. *IETF Internet Draft*, April 2003. draft-park-fasthandover-agent-fmipv6-00.

[67] C. Perkins. Mobile IP. *IEEE Communications Magazine*, (5):84–86, 1997.

[68] P. Reinbold and O. Bonaventure. A Survey of IP Micro-mobility Protocols. 2002. [Online] Available: `http://citeseer.ist.psu.edu/reinbold02survey.html`.

[69] J. Satish. L2 Considerations for Optimized IP Mobility. *IETF Internet Draft*, July 2003. draft-satish-l2-mobilereq-01.

[70] S. Sharma, N. Zhu, and T. Chiueh. Low-Latency Mobile IP Handoff for Infrastructure-Mode Wireless LANs. *IEEE Journal on Selected Areas in Communications*, 22(4), May 2004.

[71] M. Shin, A. Mishra, and W. Arbaugh. Improving the Latency of 802.11 Handoffs using Neighbour Graphs. *International Conference On Mobile Systems, Applications And Services (MobiSys)*, pages 70–83, June 2004.

[72] P. Struhsaker. Including VoIP over WLAN in a Seamless Next-Generation Wireless Environment. *white paper*, June 2003.

[73] L. Sun, G. Wade, B. Lines, and E. Ifeachor. Impact of Packet Loss Location on Perceived Speech Quality. *Proceedings of 2nd IP-Telephony Workshop (IPTEL '01)*, pages 114–122, April 2001.

[74] Agere Systems. IEEE 802.11 Channel Seleciton Guidelines. *Orinoco Technical Bulletin 003/A*, November 1998.

[75] P. Tan. Recommendations for Achieving Seamless IPv6 Handover in IEEE 802.11 Networks. *IETF Internet Draft*, February 2003. draft-paultan-seamless-ipv6-handoff-802-00.

[76] J. Vatn. An Experimental Study of IEEE 802.11b Handover Performance and its Effect on Voice Traffic. Technical report, KTH, Royal Institute of Technology, 2003.

[77] A. Weyland. Mobile-Controlled Handover in Wireless LANs. Master's thesis, University of Bern, 2001.

[78] Y. Xiao. QoS Provisioning at the MAC Layer. *IEEE Wireless Communications*, pages 72–79, June 2004.

[79] A. Yegin. Link-layer Triggers Protocol. *IETF Internet Draft*, June 2002. draft-yegin-l2-triggers-00.

Appendix A

Mobile IPv6 Movement Detection

The following appendix describes the Mobile IPv6 movement detection process. IPv6 movement detection relies on slightly different mechanisms as those used in IPv4. A brief overview of the main differences between IPv6 and IPv4 movement detection will be provided.

Mobile IPv6 movement detection consists of the following three stages [20]:

1. Initially, the MN receives an indication that movement to a new IP network *may* have occurred. This indication may arise as a result of several factors and is not always conclusive.
2. The MN probes the current access router, confirming bi-directional reachability and the validity of its CoA.
3. Once movement has been confirmed, a new access router (AR) is discovered and selected using Router Discovery messages.

When a MN has received information necessary for configuring its IP parameters, movement detection ends. The MN subsequently configures its CoA, as described in Chapter 3.

Generic network-based movement detection uses Neighbour Unreachability Detection (NUD) to detect when the default access router is no longer reachable. A MN that wishes to send IPv6 packets to another network uses NUD to test if the current default access router is bi-directionally (symmetrically) reachable [51]. This can be determined in two ways. An access router is reachable if the mobile node's upper-layer connections are making "forward

progress". For example, a TCP acknowledgement is an indication that bidirectional communications are taking place. If such upper-layer indications are not available, the reception of a neighbour advertisement in response to a solicitation also confirms bi-directional reachability [61].

NUD can quickly ascertain that a router is *reachable* by sending a neighbour solicitation and immediately receiving a neighbour advertisement in reply. However, determining that a neighbour is *unreachable* is more difficult and time consuming. If a response does not arrive within one second of sending a neighbour solicitation, a MN will retransmit the solicitation. A MN will transmit up to three solicitations (with corresponding timeouts) bMobileefore confirming that the router is unreachable. It thus takes 3 seconds to determine that a router is no longer on the link [20].

In order to minimise signalling overhead caused by NS and NA messages, NUD is only used by a MN when it has packets to send. Therefore in addition to NUD, the MN can use the Router Advertisement Interval option (when available) to detect a network level handoff. If a scheduled advertisement broadcast from the current router does not arrive within the Advertisement Interval, then the MN assumes that it has been missed. The MN will conclude that it has performed a network level handoff after missing a certain number of these advertisements. A MN is free to determine how many router advertisements it is willing to miss. This algorithm is similar to the lifetime-based algorithm used in Mobile IPv4.

When low advertisement rates are used, both the NUD and Advertisement Interval based mechanisms can be supplemented with other movement detection techniques. For example, a MN can use indications or information from the link layer to determine whether IP movement may have occurred. These indications must be verified in the second stage of movement detection as a link layer handoff does not necessarily imply layer 3 movement. Lastly, the eager-binding policy introduced within the context of Mobile IPv4 can also be used in Mobile IPv6 [20].

In summary, indications of network-layer movement can be generated usingThe Mobile IPv6 movement detection process is somewhat different to Mobile IPv4. a combination of the following mechanisms:

- Neighbour Unreachability Detection
- Advertisement Interval-based timeout

- Link layer indication
- Eager-binding

A neighbour solicitation/advertisement exchange is always performed using link-local IPv6 addresses. Link-local addresses only have local scope, and therefore can not uniquely identify a router or the current network. For example, a router could (theoretically) use the same link-local address on more than one interface [51]. For this reason, after a receiving a movement indication, the MN must verify that its AR is reachable and that its CoA is still valid (stage 2). Verifying AR bi-directional reachability is only necessary in this stage when it has *not* been performed as part of the initial indication (stage 1), such as after an advertisement interval timeout or link layer indication. These verifications are important because they ensure that unnecessary handoffs are avoided when the current AR is reachable. The Mobile IPv6 specification [51] does not describe the exact method used to confirm these parameters.

To test AR reachability (if necessary), the MN can perform NUD using NS/NA messages. Alternatively, instead of performing 3 consecutive broadcasts, this can be done by sending a single NS and waiting for a reply until a timeout has occurred [51]. After this, a RS/RA exchange must take place, irrespective of whether or not the AR is reachable. The returned router advertisement will allow the MN to verify that it is on same network and that its CoA is valid. When movement has occurred, this exchange also allows new routers to be discovered. A RS/RA exchange includes a number of built-in delays (for the same reasons described in ICMP Router Discovery above). A MN should wait for a random interval (0–1 second) before transmitting a router solicitation. Likewise, a router must include a random delay (0–500 ms) before replying with a RA [20].

Unfortunately, both NS/NA and RS/RA exchanges must take place. A neighbour advertisement is always sent in reply to a solicitation, and thus may be used to confirm reachability. However, NS messages do not contain IP information that can be used to identify the router or network. On the other hand, a router advertisement can include this information, but does not carry an indication that it was solicited by a NS. RS/RA messages therefore do not confirm reachability [20].

The third step in movement detection, where a new AR is detected if movement has occurred, usually overlaps the second step. Any RA (whether solicited in stage 2 or broadcast) will allow the MN to discover a new access router. When this takes place, movement detection has been completed.

Appendix B

VoIP Over Wireless LAN

This section investigates the extent to which different 802.11 mechanisms support time-sensitive applications such as VoIP.

802.11 networks are able to support the relatively modest bandwidth requirements that VoIP connections impose. A VoIP stream will typically use about 64 kbps using a standard PCM codec (G7.11). More sophisticated codecs can reduce this bandwidth requirement to under 10 kbps. Obviously, different 802.11 physical implementations are able to support different numbers of concurrent calls because of the various data rates they support. However, an in depth study into the capacity of 802.11 wireless LANs is beyond the scope of this document. Detailed 802.11 capacity analyses are performed in [44, 72] that take into account factors such as number of users and types of audio codecs.

The PCF allows 802.11 wireless LANs to provide rudimentary support for real-time applications. As explained previously, 802.11 access points control access to the wireless medium by polling their associated stations. However, the 802.11 standard does not specify the exact polling strategy. By incorporating a suitable scheduling scheme, an AP can ensure that a station experiences reduced medium access delays[1]. Contentionless access is also catered for by the PCF. Furthermore, the variance in the medium access is minimised because stations are given access to the wireless medium at relatively fixed intervals [53]. These factors are especially important when considering VoIP systems. Aside from the bandwidth that PCF tries to assure, low medium access delays will help to minimise the total end-to-end delay. Low access delay variations are also important because this minimises the jitter introduced by medium access mechanisms.

[1] as compared to DCF operation.

Despite these factors, PCF has significant limitations and has not been implemented by wireless equipment manufacturers. This is mainly due to its complexity and inefficiency when transporting data. In addition, PCF does not provide complete QoS support, as it does not provide an admission control function to regulate access to the wireless medium. This means that high traffic levels may degrade the service experienced by all stations [78].

DCF is even less well suited for supporting VoIP traffic. The medium access delays are very high at low data rates (e.g. 100 ms at 2 Mbps). Although this phenomenon is reduced at higher rates, the DCF does not offer low variance in the medium access time. The reason for this is that a station is not guaranteed access at a specific interval, but rather has to wait until the medium is free. This means that audio traffic may experience unsatisfactory delays and delay variations. Performance will worsen as a greater number of stations compete for the same medium [53].

Wireless LANs (in their current form) face a number of challenges when supporting real-time applications such as VoIP. Aside from the issues raised above, it has been shown in Chapter 2 that 802.11 handoffs introduce large and highly variable delays in a host's connectivity. All or these shortcomings are highlighted within network architectures such as the ones assumed in this study (refer to Chapter 3). Further research in these areas is essential if future IP-based wireless access networks are to provide these types of services.

Appendix C

Additional Tables and Calculations

This section contains additional information on certain aspects that were briefly touched upon in the main chapters. This includes information on the different thresholds used by the Orinoco 802.11 devices in the evaluation framework. Recommendations used to select AP channels within a wireless LAN have also been included. Lastly, a derivation of the equations discussed in Chapter 3 (section 3.6.1) is provided.

C.1 Orinoco 802.11 Thresholds

The Table C.1 is specified within a technical bulletin published by Agere Systems [4]. It was used to ascertain the cell search threshold used by the hybrid system. This threshold determines when an 802.11 card will begin searching for new access points and thus forms the first stage of an 802.11 handoff. It was discovered that the "low" AP density thresholds were used by the devices within the evaluation framework despite the proximity of all APs.

C.2 802.11 Channel Selection Guide

Table C.2 was published within another Agere Systems technical bulletin [74]. This table outlines the minimum channel separation that must be included when AP coverage areas overlap (e.g. between AP A and AP B). All channel combinations that ensure these radio coverage areas are sufficiently isolated are marked with a $\surd$.

	AP Density		
Threshold	Low	Medium	High
Carrier Detect (dBm)	-95	-90	-85
Defer (dBm)	-95	-85	-75
Cell Search (dB)	10	23	30
Out of Range (dB)	2	7	12
Delta SNR (dB)	6	7	8

Table C.1: Orinoco WaveLAN thresholds

	AP B channel #												
AP A channel #	1	2	3	4	5	6	7	8	9	10	11	12	13
1						√	√	√	√	√	√	√	√
2							√	√	√	√	√	√	√
3								√	√	√	√	√	√
4									√	√	√	√	√
5										√	√	√	√
6	√										√	√	√
7	√	√										√	√
8	√	√	√										√
9	√	√	√	√									
10	√	√	√	√	√								
11	√	√	√	√	√	√							
12	√	√	√	√	√	√	√						
13	√	√	√	√	√	√	√	√					

Table C.2: 802.11 channel selection guide

C.3 Calculations

On average, a mobile node will arrive on a new link midway between two periodic agent advertisements. If the agent advertisement interval varies randomly between MAXINTERVAL and MININTERVAL, the average time until a new advertisement is received after a mobile node arrives on a link (AGENTDISCOVERY) is defined by the following equation:

$$AgentDiscovery = \frac{(AverageInterval)}{2}$$

$$\therefore AgentDiscovery = \frac{\left(\frac{MaxInterval+MinInterval}{2}\right)}{2}$$

$$\therefore AgentDiscovery = \frac{MaxInterval + MinInterval}{4}$$

A mobile node generally leaves a particular network halfway through its advertisement interval. In addition, the advertisement lifetime expires when the MaxInterval time has elapsed since the last received advertisement, assuming that the lifetime spans only one interval. The average time until the advertisement lifetime expires after a mobile node leaves a particular IP network is called the residual lifetime (RL). The following equation defines the average residual lifetime:

$$RL = \frac{AverageInterval}{2} + (MaxInterval - AverageInterval)$$

$$\therefore RL = \left(\frac{MaxInterval + MinInterval}{2}\right) + MaxInterval - \left(\frac{MaxInterval + MinInterval}{2}\right)$$

$$\therefore RL = 0.75 \times MaxInterval - 0.25 \times MinInterval$$

Usually the advertisement interval spans more that one interval. For example, the advertisement lifetime may expire after 3 intervals have passed with no received advertisements. In this case the residual lifetime is:

$$RL = (0.75 + (n-1)) \times MaxInterval - 0.25 \times MinInterval$$

where n is the number of advertisement intervals.

Appendix D

HermesAP and Wireless Card Configuration

D.1 Introduction

This appendix provides an informal set of directions for configuring an Agere Systems Orinoco wireless LAN PCMCIA card under the Linux operating system. A basic understanding of the Linux fundamentals, such as compiling/installing both package and Linux kernel source code, is assumed and many steps are described very briefly.

The drivers for a PCMCIA Orinoco wireless card are included in the pcmcia-cs package (available from `http://pcmcia-cs.sourceforge.net`). The pcmcia-cs package installation consists of two stages. If the current Linux kernel does NOT include the pcmcia drivers (either as part of the kernel or as modules), then the pcmcia drivers are built as modules. The pcmcia-cs package also builds the user applications used to control the card (*cardctl*). In order to patch or modify any drivers included in the pcmcia package (such as installing HermesAP or enabling scan/monitor mode), kernel support for pcmcia must be DISABLED. Once the drivers are patched/modified, they can be compiled and installed as modules. This is described in Method 2 below. It is important to note that the HermesAP package allows tertiary firmware to be loaded into the RAM of the wireless card which makes the card behave like an 802.11 access point. In order to install HermesAP (which involves changing the drivers), Method 2 should be followed.

Note on kernel versions (2.4.X) To ensure the most functionality from a particular

card, the highest version of wireless extensions should be used. Jean Tourilhes' site (`http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html`) contains information on the version included in each kernel and patches are available on his site. For example, to get Wireless Extensions v16 to work on a 2.4.20 kernel, the kernel sources must be patched twice.

D.2 Method 1

Pre-install requirements:

Update Firmware (can get updates from Orinoco website)

PCMCIA Package (3.1.19 to 3.2.X)

Latest 2.4 Kernel - These kernels include support for PCMCIA drivers

Use the Orinoco_cs driver

Recompile the Kernel

make menuconfig (config, xconfig)

Enable the following options:

1. Your Ethernet network card driver
2. Hermes chipset 802.11b support (Orinoco/Prism/Symbol)
3. Hermes PCMCIA Card Support
4. AT&T WaveLAN & DEC RoamAbout Support
5. PCMCIA CardBus (not other options for old ISA bridges etc...)

This builds in the PCMCIA drivers into the Kernel. The utilities (*cardmgr* & *cardctl*) still need to be built using the pcmcia-cs package above or distribution specific package (e.g. *apt-get* on Debian). See below.

Make sure you have the right processor, your network card etc...

Recompile:

make dep clean modules modules_install bzImage

configure /etc/lilo.conf and install *lilo*

Install the PCMCIA package

I didn't do an apt-get (Debian) but downloaded the latest pcmcia-cs tarball, extracted & compiled it. The Pcmcia package includes Wireless Extension (see below) support (3.1.15 onward) It also includes orinoco drivers (3.2.4 includes orinoco_cs driver v0.13b). Read the included PCMCIA.Howto for further installation instructions.

Abbreviated: *make* config, *make* all, *make* install

Configuration files go to /etc/pcmcia/

Check that the following lines appear in the config files (Should be there automatically):

```
card Lucent technologies WaveLAN/IEEE Adapter
version Lucent Technologies, WaveLAN/IEEE
bind orinoco_cs
```

Configure /etc/pcmcia/wireless.opts, /etc/pcmcia/network.opts and /etc/pcmcia/config.opts (Shouldn't really need to do much)

The hermes.conf file from Jean Tourilhes site must be placed in /etc/pcmcia/ (`http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Orinoco.html`)

The card should now be able to be stopped and started (with some luck) without problems. If a double high-pitched beep is sounded when the pcmcia services are started, then card is configured.

/etc/rc.d/init.d/pcmcia restart (Red Hat) or /etc/init.d/pcmcia restart (Debian)

For Debian, the following lines should be inserted in /etc/network/interfaces for easy IP configuration:

```
iface eth0 inet static
address 192.168.1.2
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
```

Install Wireless Tools

Wireless Extensions (WE) is a API implemented in the kernel and drivers that allow various tools to set various wireless parameters and query various wireless statistics. Wireless Tools (WT) is a reference implementation of a set of tools to access and manipulate Wireless Extensions.

When using wireless tools 26, *make* and *make install* will compile and install the sources correctly.

For further help on the Wireless-tools package, consult the package maintainer's website: `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html`

D.3 Method 2

This method can be used to install and configure HermesAP. It is effectively the same procedure as above, but is necessary to install the modified HermesAP wireless card drivers.

Preinstall requirements:

Latest Kernel 2.4, pcmcia-cs and wireless-tools packages.

The *wget* and *unzip* programs are also needed for installing HermesAP.

Kernel Configuration

make menuconfig

1. Deactivate < > PCMCIA/CardBus support
2. Activate [*] Wireless LAN (non-hamradio) but don't select any of the card modules there. They will be compiled with the pcmcia-cs package

HermesAP

1. Select <*> /dev file system support (EXPERIMENTAL) (in file systems)

2. Select [*] Automatically mount on boot

3. Select [*] Debug devfs. This activates devfs support in the kernel. You need to install devfsd before you do this!

Recompile and install kernel. The kernel should be booted and running before the next steps are attempted.

PCMCIA Installation

For ordinary installation: decompress and install as in Method 1

For HermesAp:

Decompress pcmcia-cs package (e.g. /usr/src/pcmcia-cs-3.2.4)

Decompress hermesap package

cd to ..../hermesap-0.2/driver/orinoco-0.13c

Delete the Makefile. These drivers will be compiled within the pcmcia package.

Copy contents of orinoco-0.13c directory to the "..../pcmcia-cs-3.2.4/wireless" directory. Now the PCMCIA package supports HermesAP and can be compiled (as in Method 1)

Restart the machine. It seems this can prevent problems with mounting the network interface (usually eth[x]) on the devfs filesystem.

Compile Wireless tools as above

HermesAP Install

The HermesAP package contains patches for orinoco-0.13c and pcmcia-cs-3.2.3. If these versions are not used in the pcmcia package, then all the .c and .h files from driver/orinoco-0.13c/ must be copied over the ones in the "kernel/pcmcia_cs-x.x.x/wireless". The pcmcia-cs package can then be compiled and installed as before.

View the ..../hermesap-0.2/docs/README. (Intel is little-endian!)

IMPORTANT: Make sure that eth[x] is *ifconfig*'ed down before loading firmware.

Troubleshooting

When loading the tertiary firmware into the card's RAM, I get the error:

```
# ./hfwload eth1 ../firmware/T1085800.hfw
can't open /dev/orinoco/eth1_mem
```

The devfs filesystem has not been compiled into the kernel. Enable devfs in kernel and install devfsd (as above)

See the HermesAP FAQ for other helpful information: `http://www.comteam.at/~alex/phpBB2/index.php`

Appendix E

Evaluation Framework Utilities

E.1 Introduction

An overview of the hardware and software components that make up the evaluation framework were presented in Chapter 5. This section will provide some additional information on the software systems that were developed to evaluate Mobile IP handoffs. A description is also given of the Robust Audio Tool (RAT) that was used to perform the VoIP analysis discussed in Chapter 6. These components of the evaluation framework are described in the hope that they may be used effectively in future projects that involve these technologies. Some of these techniques may even be useful in projects that have a completely different focus.

E.2 Mobile Node Applications and Scripts

All Mobile IP timing measurements were carried out on the mobile node. An explanation of how this was performed through a combination of user applications and Linux shell scripts is outlined below. The following utilities were developed during the course of this project specifically for the evaluation framework.

dynmnd The Dynamics mobile node daemon can be executed to output verbose debugging information. The dynmnd source code has also been modified to output additional timing information as different states are entered. This output information is

piped to a file that can be analysed once the daemon is deactivated. This can be achieved with the following operation:
`dynmnd --fg --debug > dynmnd.log`

set_policy.c The *set_policy* utility allows the current agent selection policy to modified (e.g. eager or lazy-binding). The Dynamics mobile node API provides an interface to the running dynmnd daemon. Dynamics supports the following policies:

- Early-expire
- Newest-FA
- Eager-Switching
- Newest-ADV

It was discovered that the "Eager-switching" policy did not perform exactly as specified by the Mobile IP eager-binding specification. Instead, the "Newest-ADV" policy was used to implement eager-binding. This policy executes a handoff after the first advertisement from a previously unheard agent is received. For more information on these policies, consult the dynmnd manual.

visit-fnet & go-home These shell scripts initiate an 802.11 handoff to a different access point by changing the current ESSID (`iwconfig eth0 essid [home/fnet]`). They also record the current system time using the *record_time* utility that indicates the start time of a Mobile IP handoff.

syslog_interface.c As described in Chapter 6, the *syslog_interface* receives events from the Linux kernel that indicate when an 802.11 handoff has occurred. This is achieved by monitoring the `/dev/xconsole` FIFO (with root privileges). When an event is received that signals the completion of an 802.11 handoff, the system time is recorded and logged.

solicit2.c The solicit2.c (version 2) application is an extension of the *syslog_interface* that has been specifically designed to implement the Hinted Cell Switching (HCS) mechanism. Additional functionality has been incorporated into the general syslog interface that allows a solicitation/advertisement exchange to be initiated immediately after an 802.11 handoff. This is achieved via the Dynamics mobile node API.

package.c Once a Mobile IP handoff test has been performed, the *package* tool will search through the dynmnd log file (e.g. dynmnd.log) and sift appropriate timing information. Link layer handoff initiation and completion times are also loaded. This information is compiled into a list of major Mobile IP and 802.11 events (such as "registration reply received" or "new advertisement received") along with the time interval between each event.

extract.c After several Mobile IP handoffs have been performed, each corresponding output of the *package* utility is then collated into a single file by the *extract* application. A summary of the Mobile IP handoff performance for the entire series of tests is created, with timing information on the three significant stages of Mobile IP handoff (802.11 handoff, movement detection and registration).

factl_client.c Once a handoff test has been completed, the *factl_client* program contacts the home agent and reinitialises the Dynamics home agent daemon (dynhad). For example, this is used to ensure that no previous Mobile IP tunnels exist at the start of a new test.

E.3 Robust Audio Tool (RAT)

The Robust Audio Tool (RAT) is an application that allows audio information to be streamed over a data network. Is supports a number of different audio codecs and several additional features such as error concealment. VoIP data is transmitted over the network using the RTP/UDP/IP protocols. A screen-shot of the RAT application is depicted in Figure E.1.

RAT is used to establish VoIP calls over the evaluation framework, between the mobile node and a corresponding node. The corresponding node streams audio information from a file (`ttm01.au`) to the mobile node. The mobile node in turn saves the received data to an output audio file. The information lost during a Mobile IP handoff is reflected in the output audio file by a sudden period of silence. These periods of silence range in magnitude depending on the movement detection mechanism used. In order to determine the subjective user opinion of a VoIP application's performance during a Mobile IP handoff, short samples of the output audio files were played to various listeners. The following table describes the location of these handoff disruptions within each file.

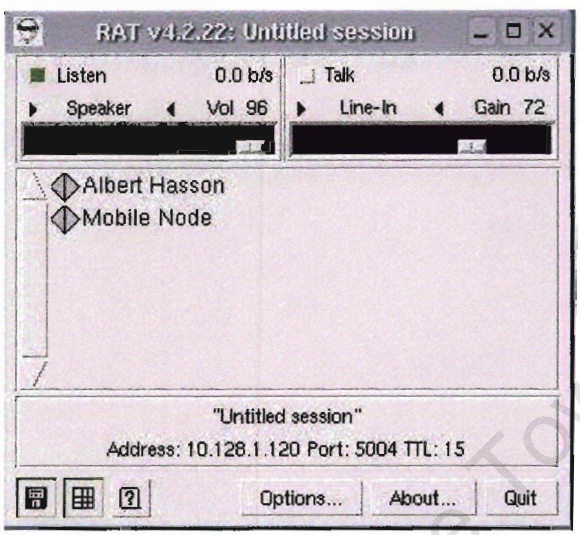

Figure E.1: Robust Audio Tool screen-shot

Movement Detection Technique	Audio File	Position (min:sec)
Lazy-binding (LCS)	voip-dwn-lcs.wav	1:10
Eager-binding (ECS)	voip-dwn-ecs.wav	0:10
Hinted Cell Switching (HCS)	voip-dwn-hcs.wav	1:18
HCS/FastADV	voip-dwn-fast.wav	1:16
Advertisement caching	voip-dwn-advert.wav	0:14

Table E.1: Handoff disruption locations

Appendix F

Accompanying CD-ROM

The following information may be found on the CD-ROM that has been included with this text:

- **Source Code**
  All source code and shell scripts that were developed can be found in the "Source-code" directory. These include working examples that demonstrate how the different APIs (Dynamics, wireless extensions and syslog) are used.
- **Dynamics Mobile IP**
  The modified version of the Dynamics Mobile IP source code (version 0.8.1) used in the evaluation framework has been included in the "dynamics-0.8.1" directory. A patch has also been included that will suitably alter the original source code.
- **VoIP Quality Evaluations**
  The VoIP output samples that were produced using the evaluation framework have been placed in the "VoIP-evaluation" directory. A copy of the listening opinion survey questionaire has also been included.
- **Research Articles and Papers**
  Electronic copies of research papers, some which are listed in the "References" section of this text, can be found in the "Research Literature" directory.
- **Thesis document**
  This document, in both postscript form and pdf form, can be found in the "Thesis" directory.